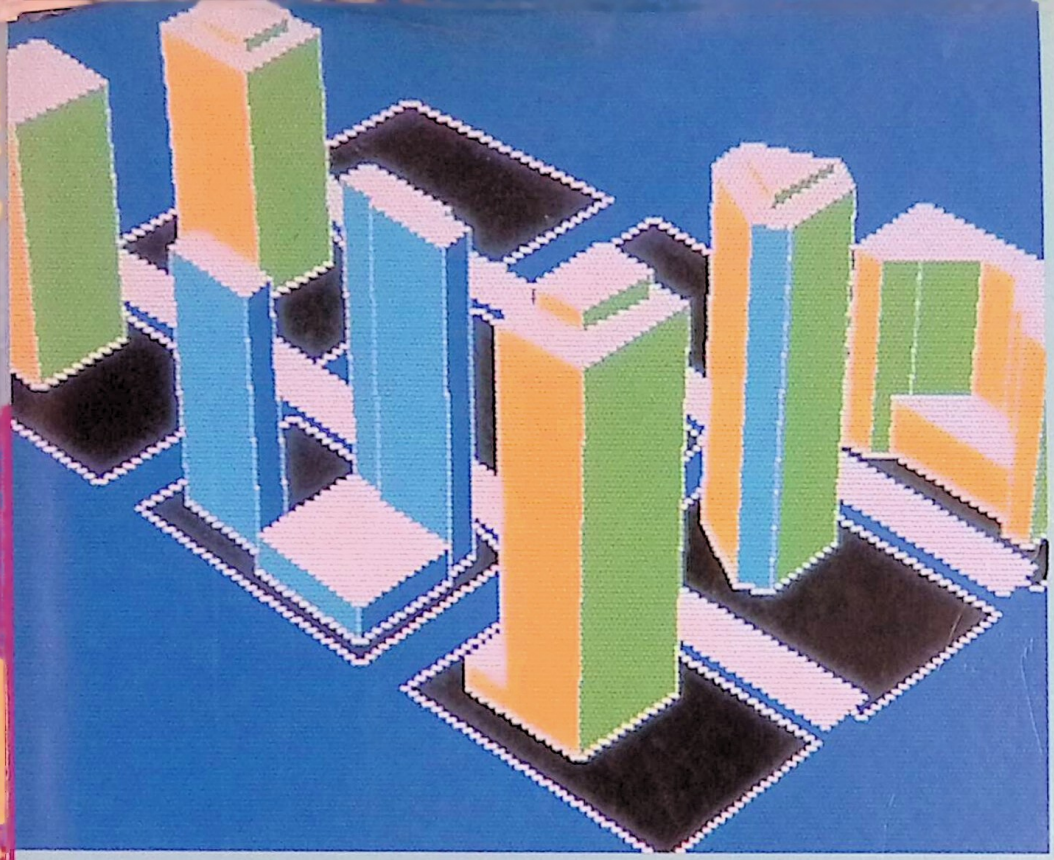
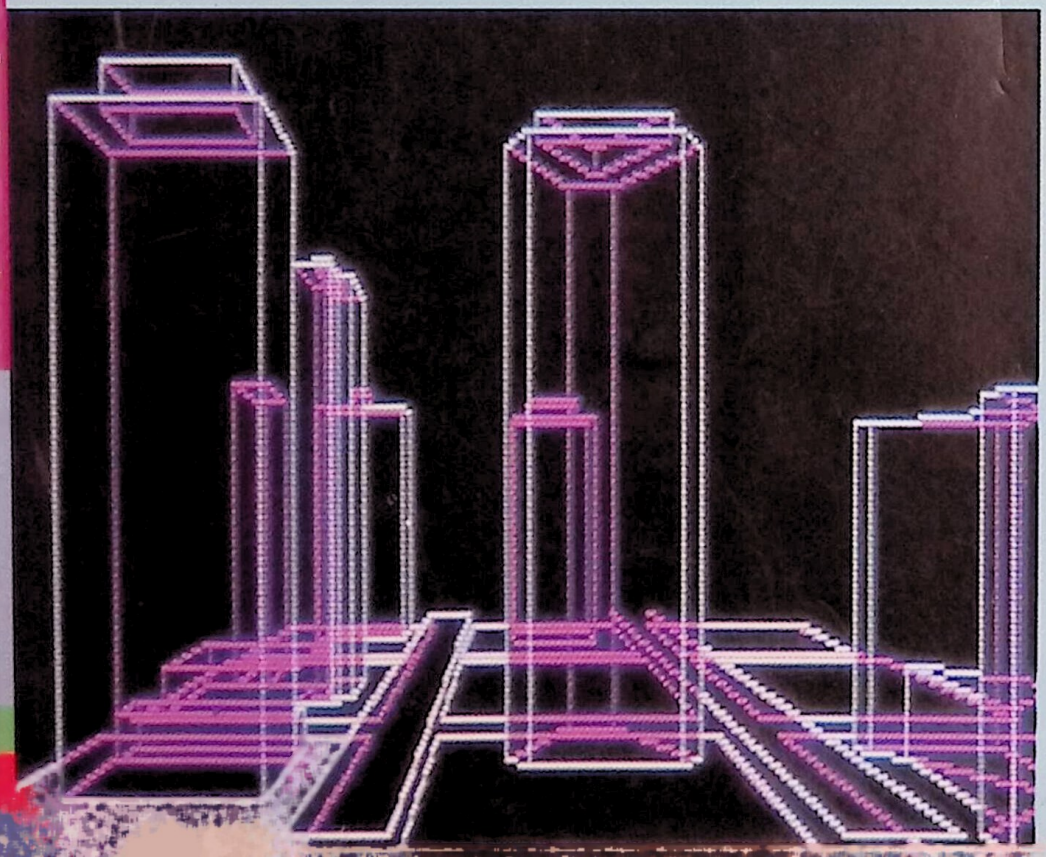


富士通
FM-7
FM NEW 7
FM77
対応

FM 実践 グラフィックス



基礎から3次元C.Gまで

田島清悟 著

山海堂

FM 実践 グラフィックス

基礎から3次元C.Gまで

田島清悟著

山海堂

まえがき

コンピュータが個人で所有できる時代となって，わずか数年の月日を経たに過ぎませんが，マイクロプロセッサ，マイコン，パソコンと呼び名が移り変わるとともに，その性能は目を見はるばかりに向上してきました。

かつてン億円もしたコンピュータとほぼ同じ性能のものが，10万円そこそこで手に入るようになってきたわけですから，これは驚異というよりなんだか愉快な気分にもなってきます。

パソコンの普及に伴い，供給されるソフトも豊富に出揃い，また刊行される専門雑誌も多種多様となって書店の店先を賑わしています。誰もが気軽にコンピュータを楽しめるようになった反面，パッケージソフトや雑誌で公開されるプログラムのほとんどがゲームで占められている現状には，少しばかり疑問が残ります。せっかくの高性能パソコン。もう少し知的な生産のための稼動を考えても良い時期かもしれません。

本書は，8ビット機として能力的に過不足のない富士通製FM-7，NEW7，77，AVを用いて，グラフィックの基礎から3次元C.Gまでを解説し，また建築やデザインなどに使用できる実用性を持たせたプログラムも掲載してみました。

第1章の写真および図は，すべて本書に掲載したプログラムで作成したものです。しばし手を休めて，パソコングラフィックスの可能性について考えて戴ければ幸いに思います。

なお，文中FM-7とあるところは，みなさんの機種に合わせてNEW-7，77，AVと置きかえてお読みください。また，本書のプログラムはF・BASIC Ver.3の環境下で動作しますので，AV使用時は，Ver.3のシステムを立ち上げてから使用して下さい。

✻✻✻✻ プログラムメディアの頒布サービスについて ✻✻✻✻

本書に収載されているプログラムのうち，スケッチボード，3DGND2，3DWLD2，3DW／HSの4本と参考データをフロッピーディスクに収録して，希望者に頒布いたしますのでご利用ください。
この頒布は入力する時間のない読者の便を図って行なうものです。下記事項をご了承のうえお申し込みください。

- ディスク自体の構造上のトラブル等による読み込み不可能な場合には，無償で交換させていただきますので，お手数ですが当社着払いでご返送ください。
- 読者が個人で使用する以外は，本プログラム内容の一部もしくは全部を山海堂に無断で，いかなる方法でも複写・複製したり，他に転載・転用することを禁じます。
- 受付けてから発送まで約1週間を要しますので予めご了承ください。

◉ お申し込み方法

- 頒価は1部につき，5インチ8000円，3.5インチ8,500円（送料は当社で負担します）。
- 下記の申し込み形式に従って必要事項を明記のうえ，現金封筒でお申し込みください。
- 申し込み先　113 東京都文京区本郷5—5—18　山海堂　営業部メディアサービス係

申し込み形式

送り先			
お名前		TEL	
必要部数	部	メディア	5インチ ： 3.5インチ
名　　称	FM-7・FM-NEW 7・FM-77	**FM実践グラフィックス**	

FM 実践グラフィックス　　— 目　次 —

1章　写真と図で綴るグラフィックス ……7

2章　基礎のグラフィックス ……19

- PSET ……19
- LINE ……21
- CONNECT ……24
- CIRCLE ……25
- PAINT ……26
- COLOR ……27
- SCREEN ……29
- SYMBOL ……30
- GET@ ……32
- PUT@ ……34
- GCURSOR ……35
- CONSOLE ……35
- CLS ……37

3章　実践プログラミング ……38

■ プログラミングのカンどころ ……38
■ グラフィック・デモンストレーション ……39
■ パソコンで絵を描こう ……41
■ 絵を描くプログラム ……42

● CONNECT 文内に直接格納 ……42
● DATA 文に格納し，READ 命令で読み出す ……43
● 配列に格納 ……44
● メモリに直接格納 ……45
● 外部記憶装置に格納 ……45

■ デザイン・シートを使おう ……45

■ スケッチボードプログラムの作成 ……47

■ グラフィックカーソル ……48
■ メインルーチン ……48
■ (1) LINE ……52
■ (2) PAINT ……52
■ (3) BRUSH ……53
■ (4) CIRCLE ……54
■ (5) SAVE ……54
■ (6) LOAD ……54
● データの構造 ……54
● 上手に絵を描くために ……56

■ 中間色の表現方法 ……58

■ 2色の組み合わせで作る中間色 ……59

■ 3色の組み合わせで作る中間色 ……61

■ 画面情報の直接転送法とスケッチボード 2 ……62

■ 転送の実験 ……63
■ マシン語の助けをかりる ……64
■ 変数の格納法を知る ……64
■ データをすり替える ……65
■ 画面転送プログラムの実際 ……67

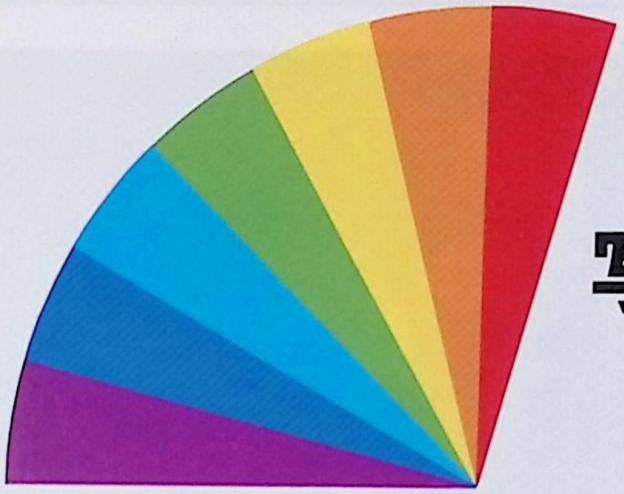
4章　3次元グラフィックス ……72

■ 画面座標系と原座標系 ……72
■ 座標系の変換（画面座標←原座標） ……73
■ 2次元図形の平行移動 ……74
■ 2次元図形の拡大および縮小 ……76
■ 図形の回転 ……76
■ 大切なクリッピング ……78
■ クリッピングの条件 ……79
■ 図形加工のサンプルプログラム ……81
■ 3次元座標 ……84
■ 3次元モデルの形状作成手法 ……85
■ 座標値と3面図との対応 ……86
■ 3次元空間の座標変換 ……88
■ 各軸まわりの回転実験 1 ……92
■ 回転実験プログラム ……96
■ サンプルデータ ……99

5章　3次元実用プログラム3題 ……102

■ 3次元地形図プログラム（3D GROUND V2） ……102
■ プログラムの実行 ……103
■ データの新規作成 ……103
■ 立体図の描画 ……105
■ ステレオ立体図 ……110
■ テスト用データ ……112

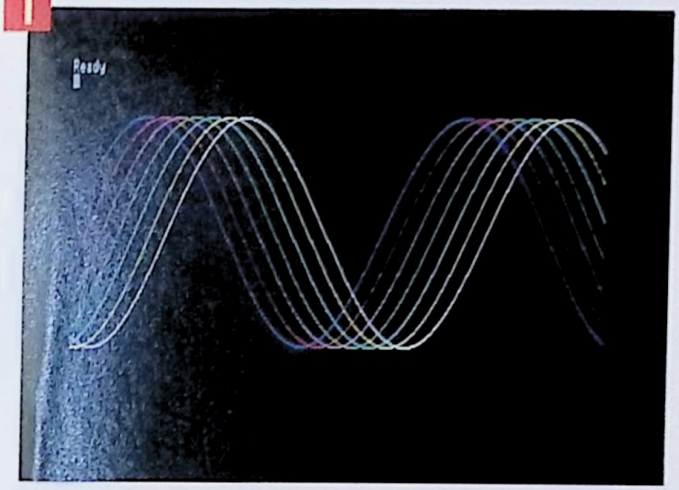
■ 3次元モデル形状作成プログラム（3D WORLD V2）……121
■ プログラムの構造と機能……121
● メインメニュー……121
● 3面図モード……123
● 省力的入力法　WALL & CIRCLE……126
■ アクションモード……131
● ディスクの無い場合の実行……135
■ 3次元高速描画バージョン（3D WORLD V2.1 Hispeed version）……150
■ プログラムの実行……150
■ アクションモード……151
■ オートアニメーションモード……152
■ プロッタモード……153
■ ローランドD.G製プロッタ使用時の変更点……154
■ マシン語プログラムの入力法……155
● マシン語モニタ……155
● マシン語の保存……157
■ ハイスピードバージョンの構成……157
■ 立体図の陰線消去法とペイント法……167



1章

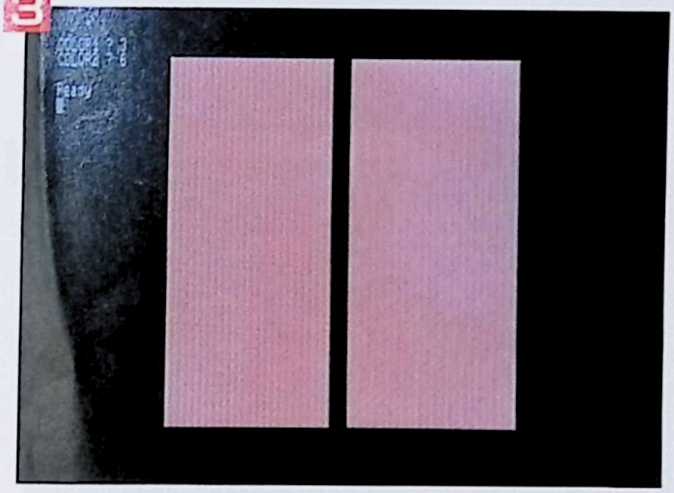
写真と図で綴る グラフィックス

1

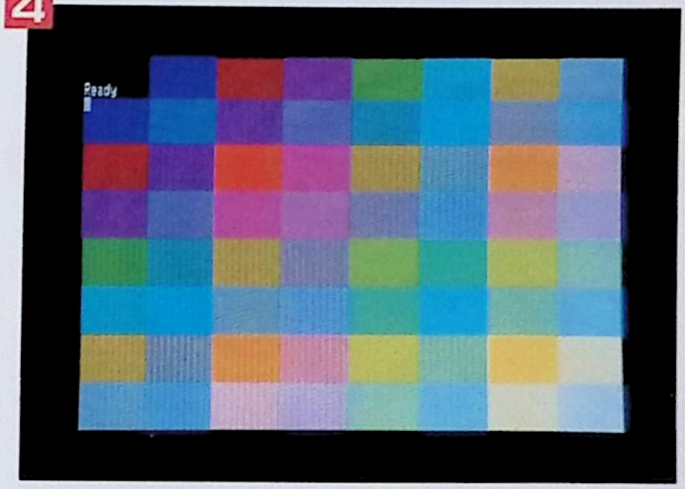


2

3

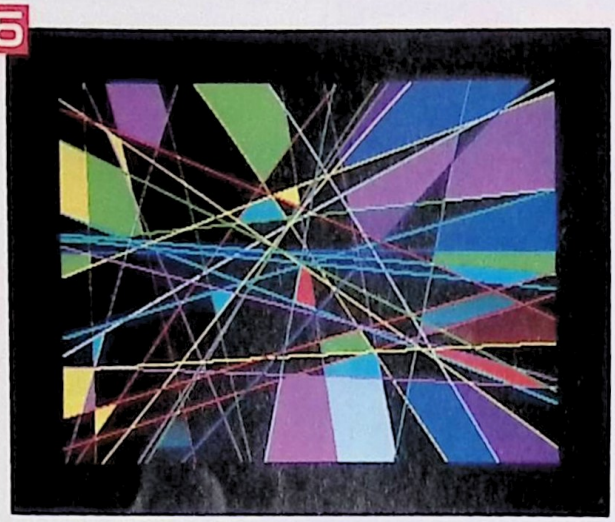


4

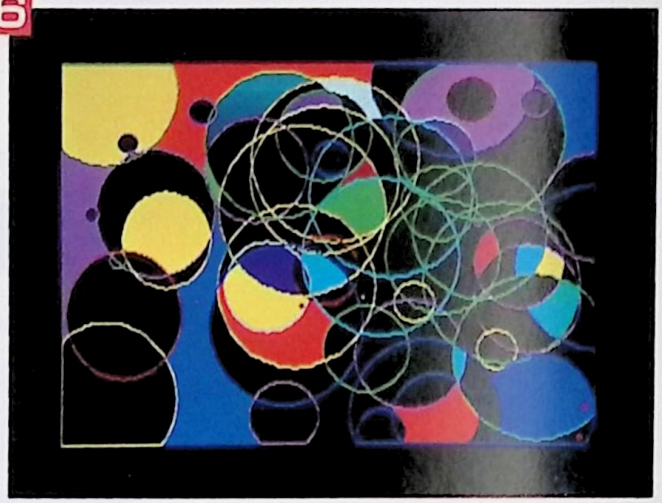


1 PSETで描いたサインカーブ（第2章21P） 2 SCREENの実験（第2章30P）
3 2色で作る中間色．左は縦，右は斜め並びのドット（第3章59P）
4 2色の組み合わせで作った中間色チャート（第3章60P）

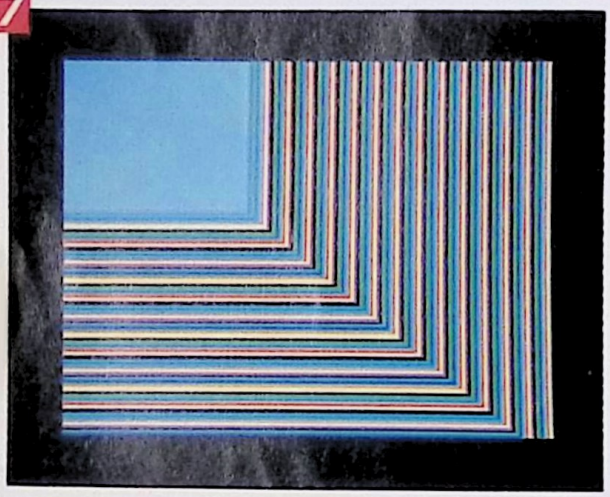
5



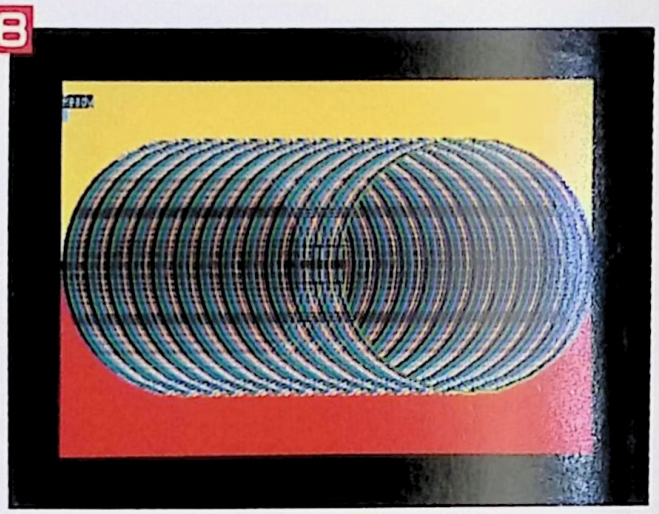
6



7

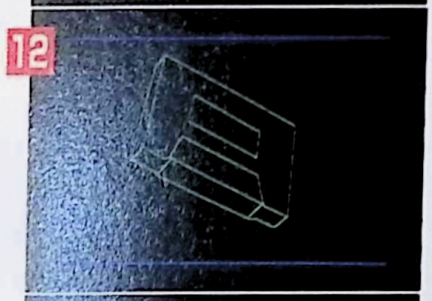


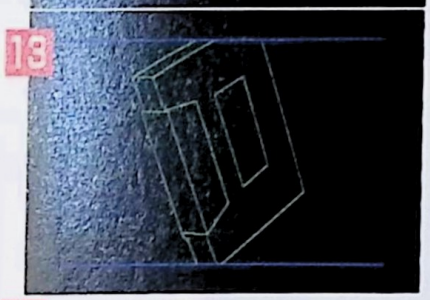
8



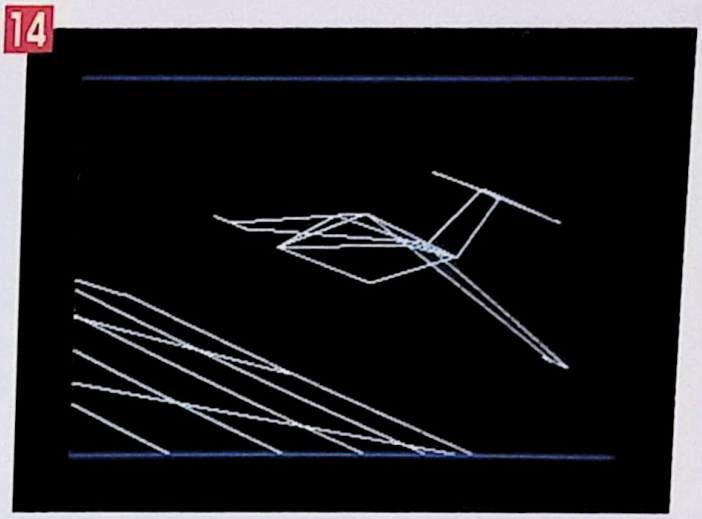
9

5 6 乱数で描いたデモ画面（第3章39P）
7 8 規則性を持たせて描いたデモ画面（第3章39P）
9 決められた手順に従ってデータを読み，アルファベットのFMを描いた実行例（第3章43P）

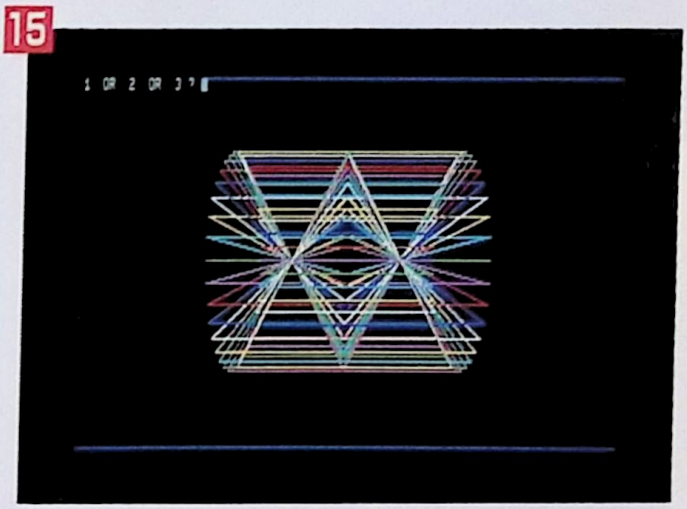


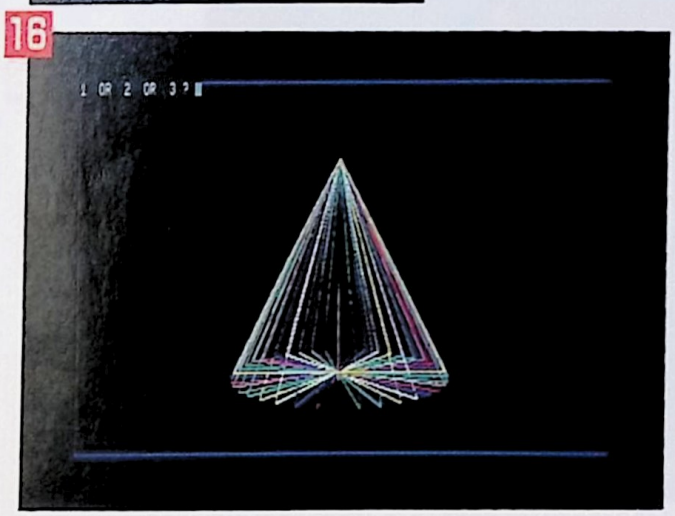


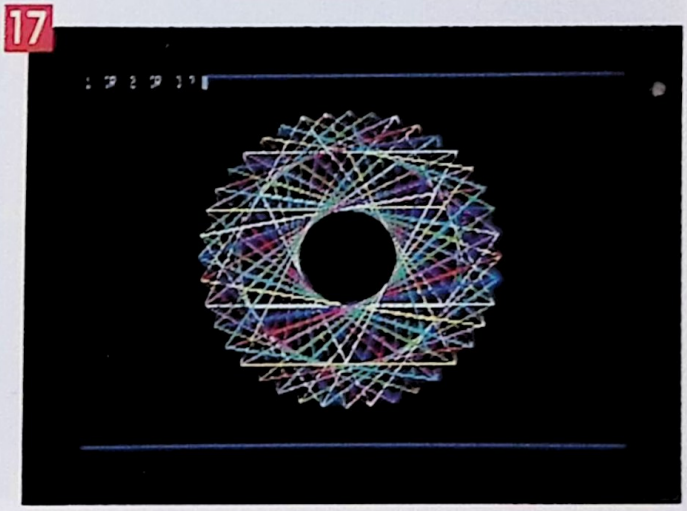
10 11 12 13
図形の 2 次元回転
（第 4 章 81P）



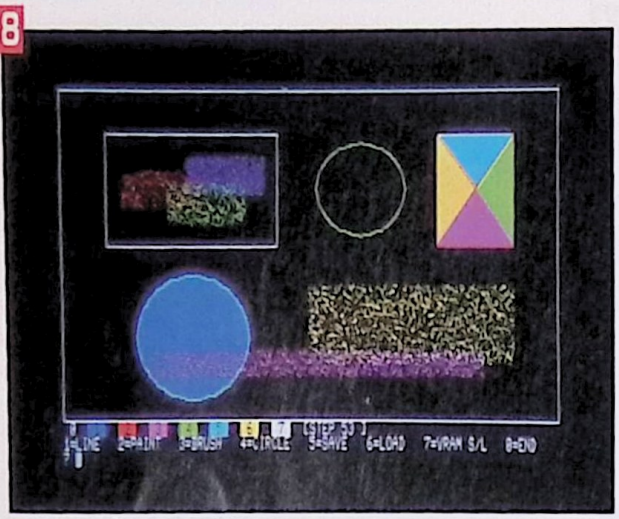
14 3 次元図形の表現（第 4 章 96P）
15 16 17　3 角板の 3 次元連続回転（第 4 章 92P）



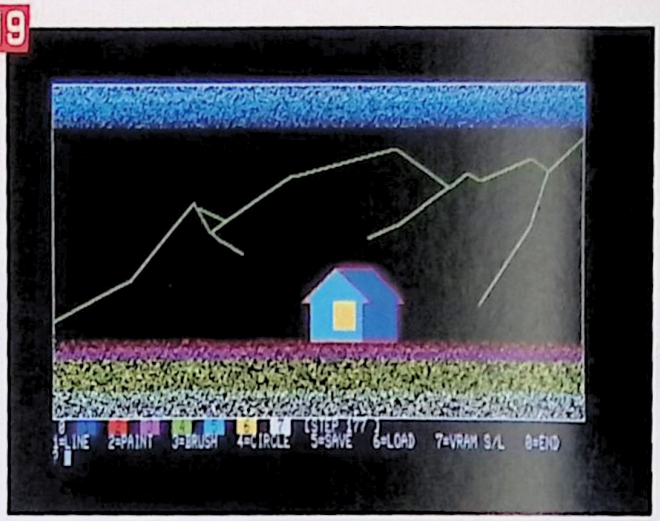




18

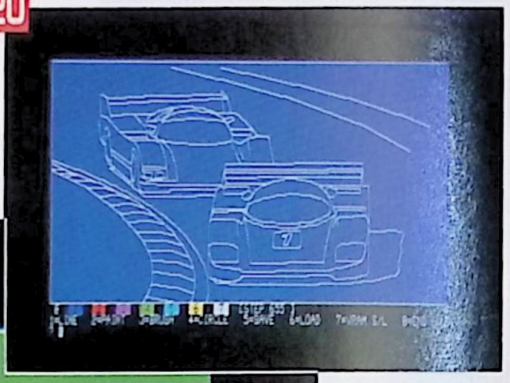


19

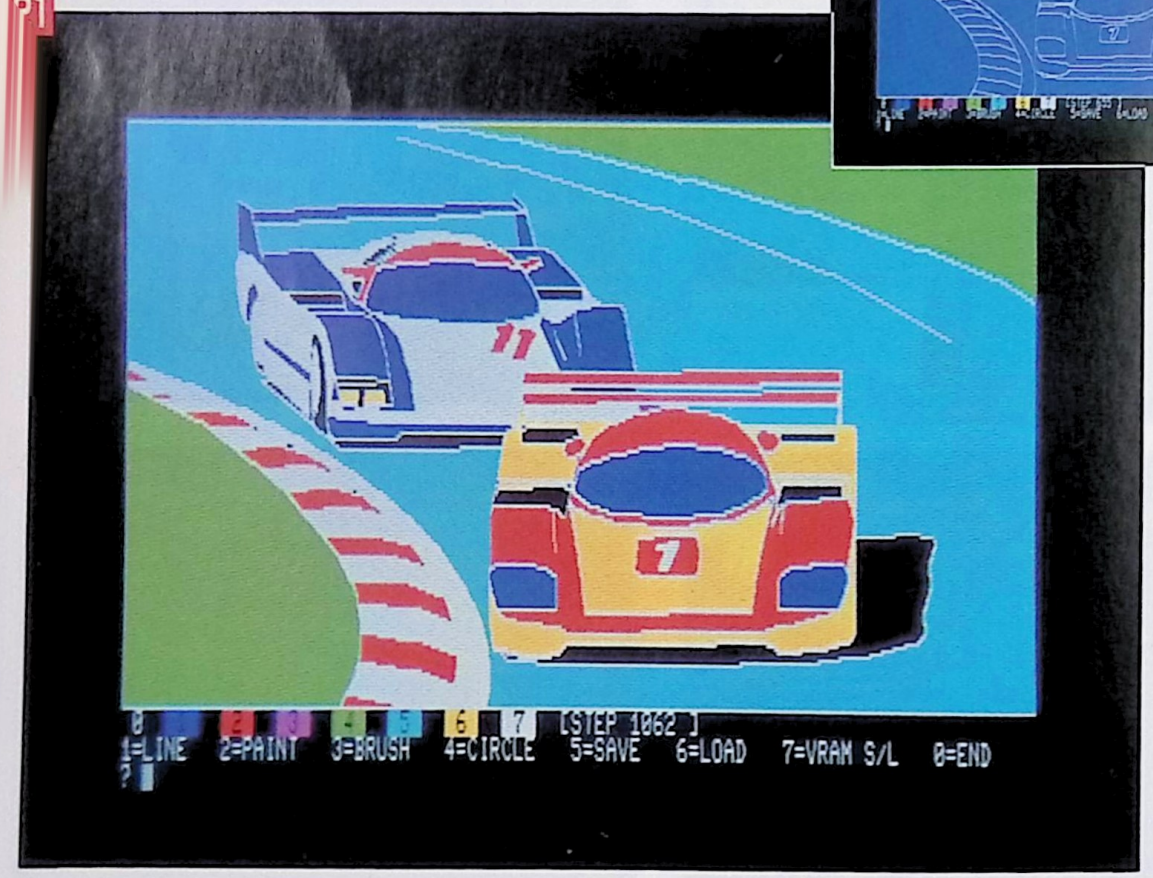


18 スケッチボードプログラムで適当に描いた絵
19 ブラシ機能を利用した風景

20



21



20 LINE機能で線画を描き，21 PAINT機能で色を塗り完成した画面

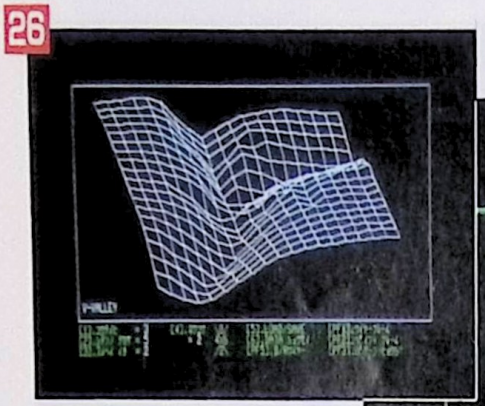
スケッチボード（第3章47P）で
描いた絵
22南極観測船「しらせ」
23舞子
24乗用車
25皇居周辺の地図

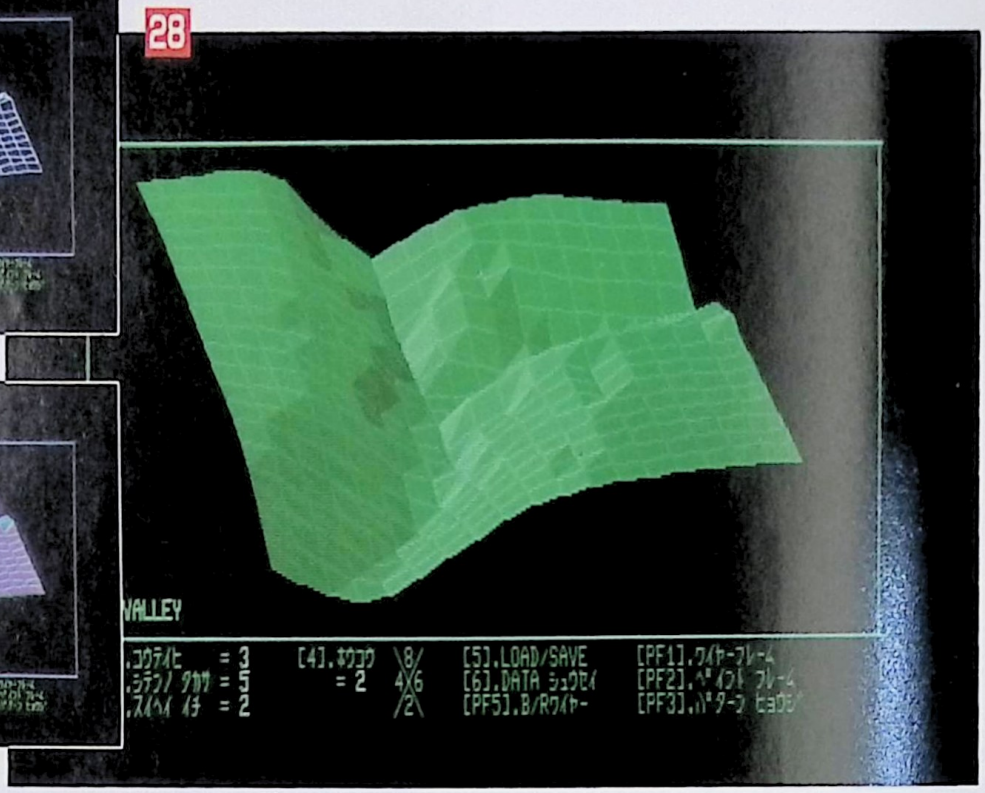
22

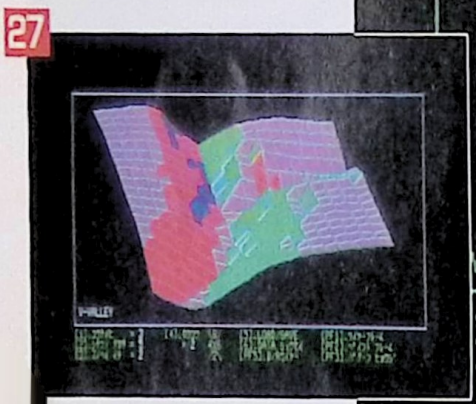
23

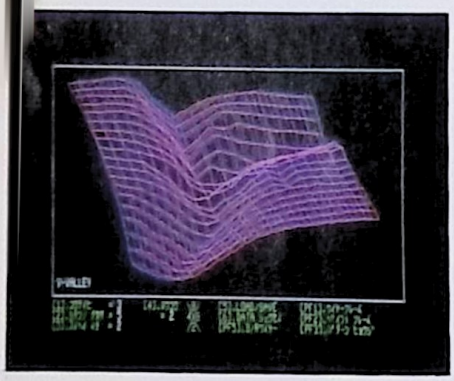
24

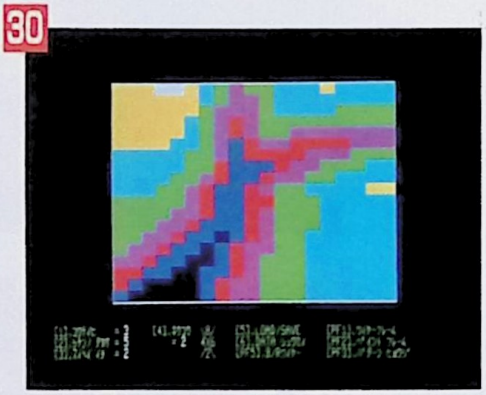
25

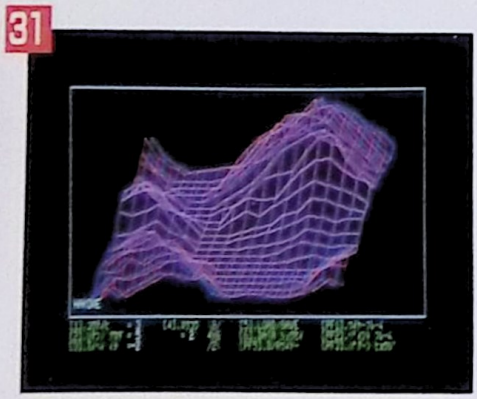


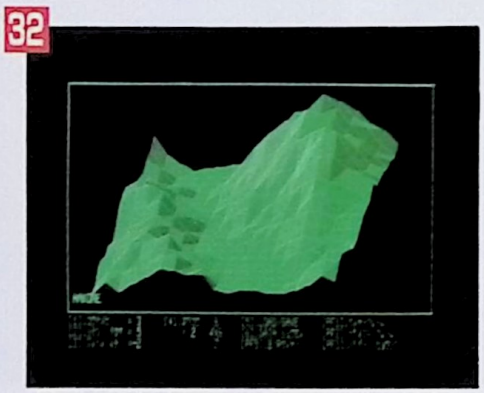












3次元地形図
（第5章102P）

26 ワイヤーによる地形表現

27 陰影表現（カラーディスプレイ）

28 陰影表現（グリーンディスプレイ）

29 ステレオ立体図

30 高度別色分け表現

31 32 箱根周辺のステレオ立体図と陰影を付けた立体図

3 次元形状作成（第 5 章121Ｐ）

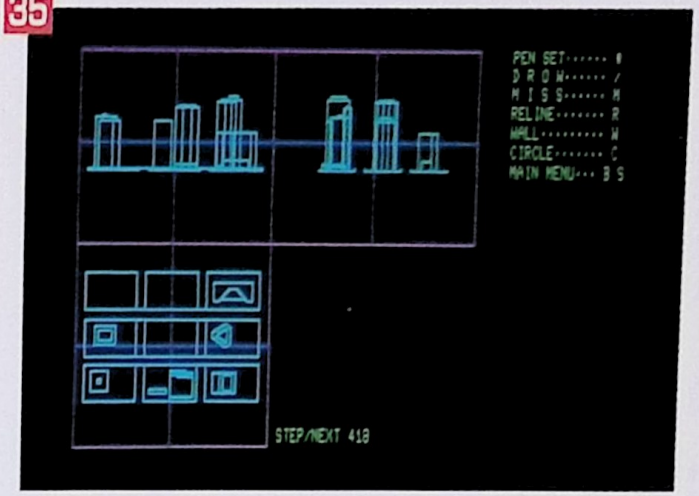
33 34副都心高層ビルの立体図生成

35対話形式によるデータ入力と副都心の３面図

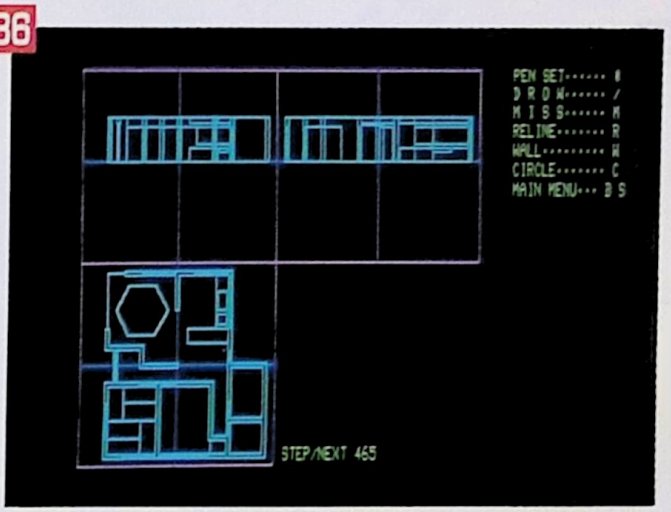
36対話形式によるデータ入力と間取りの３面図

37対話形式によるデータ入力と飛行機の３面図

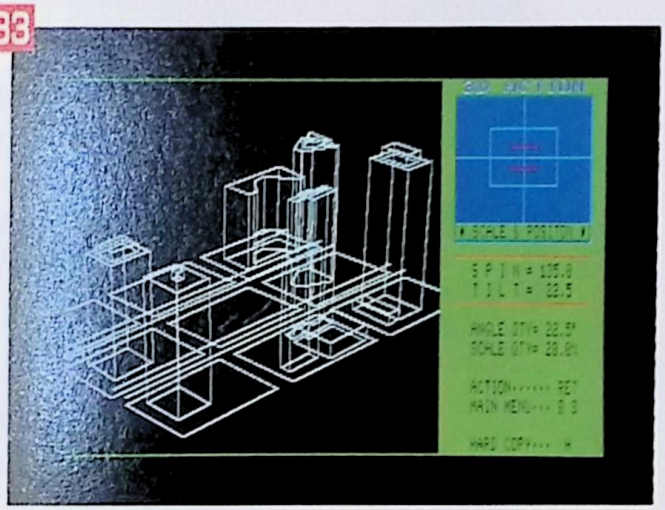
35



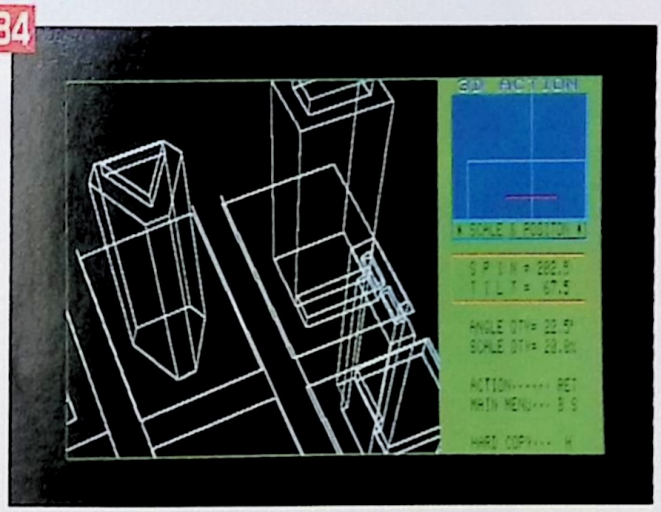
36



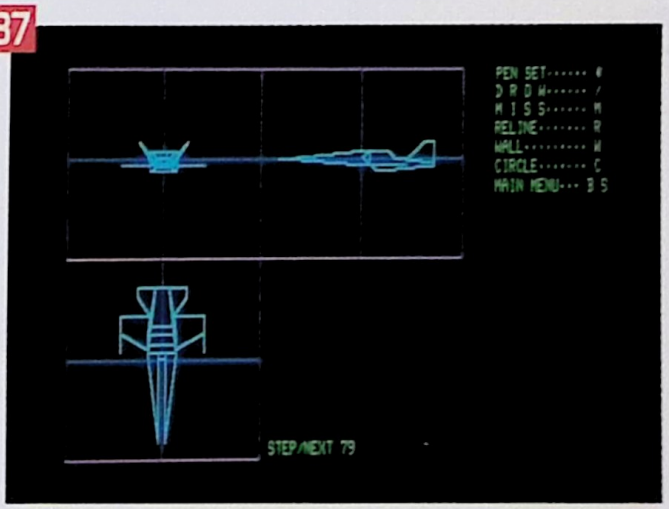
33



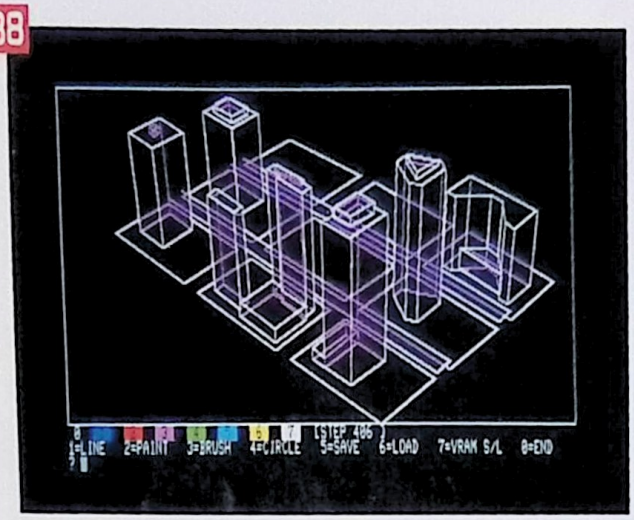
34



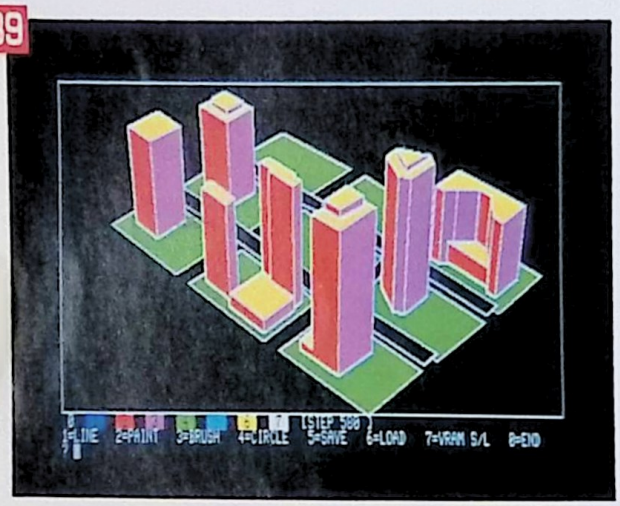
37



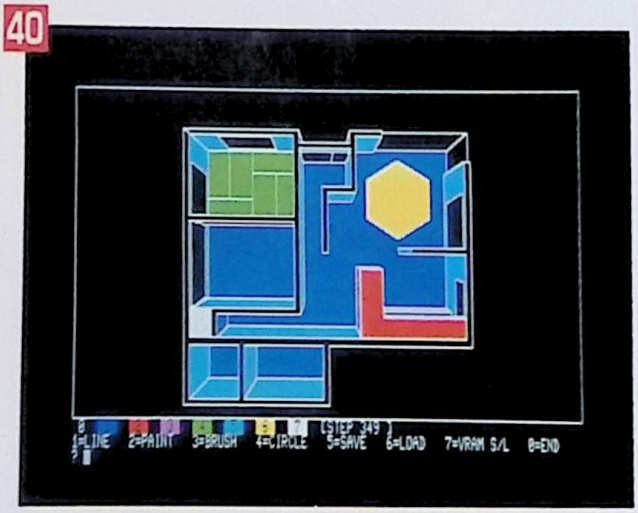
38



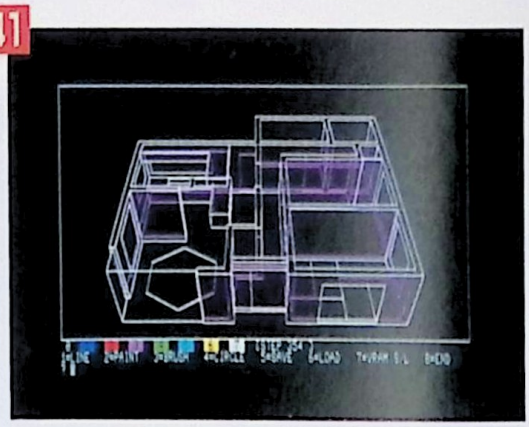
39



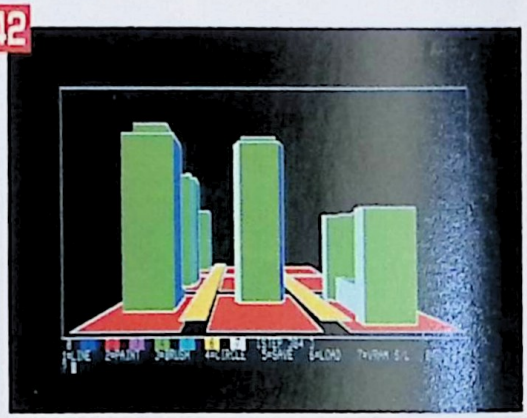
40



41

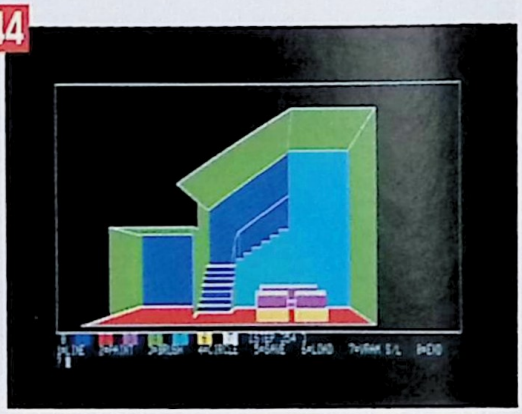


42



43

44

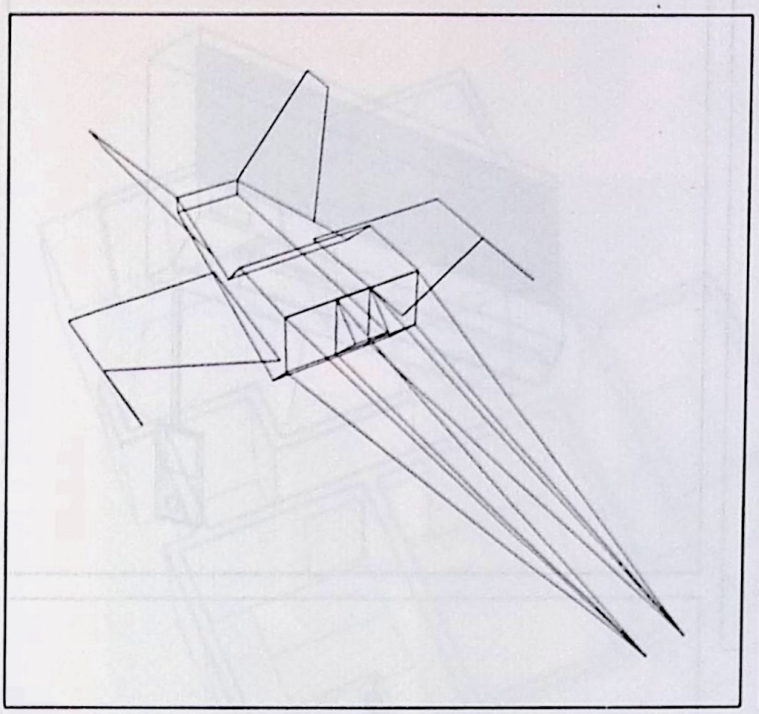


立体図のペイント（第5章167P）

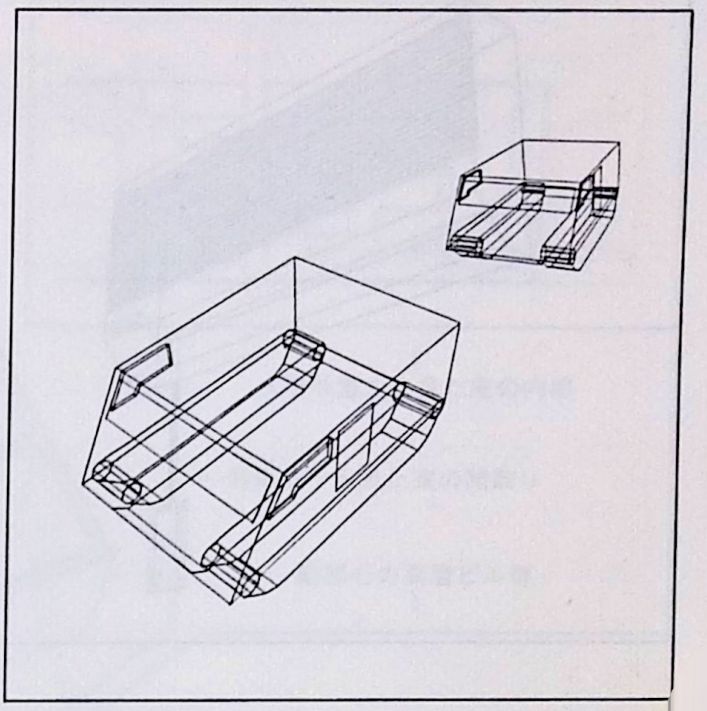
38 41 陰線処理　39 42 ペイントを実行した副都心の高層ビル

40 部屋の間取り　43 アニメ風戦闘機　44 室内の様子

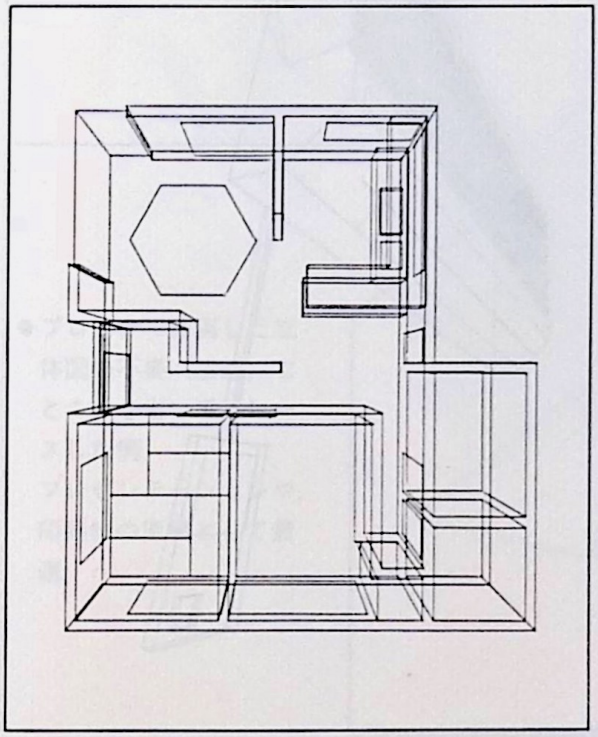
■3次元モデルのプロッタ出力例（3D　WORLD　ハイスピードバージョンプログラム）



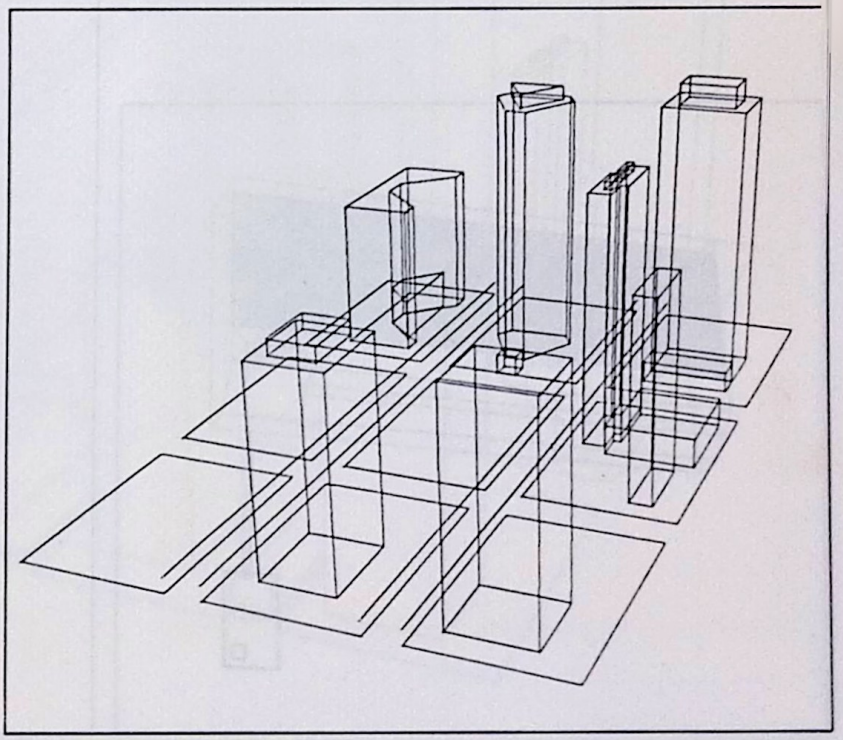
アニメ風戦闘機



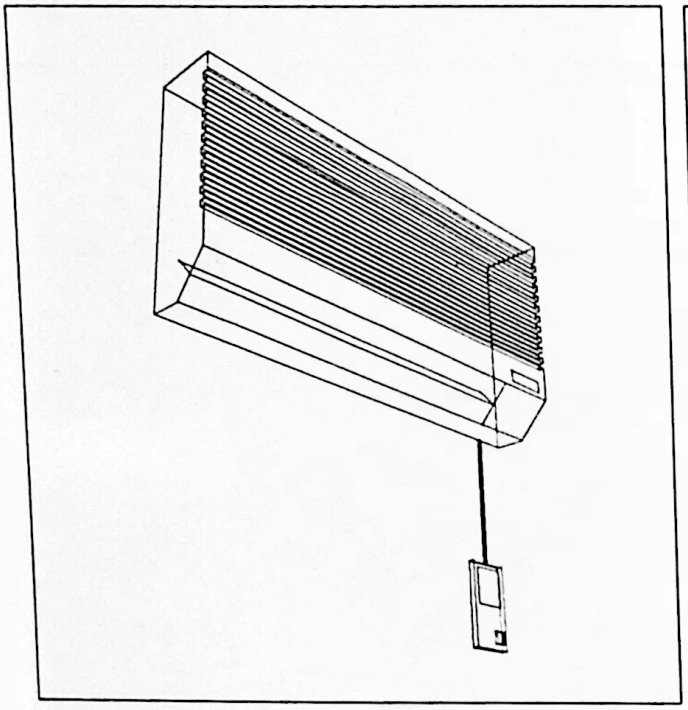
雪　上　車

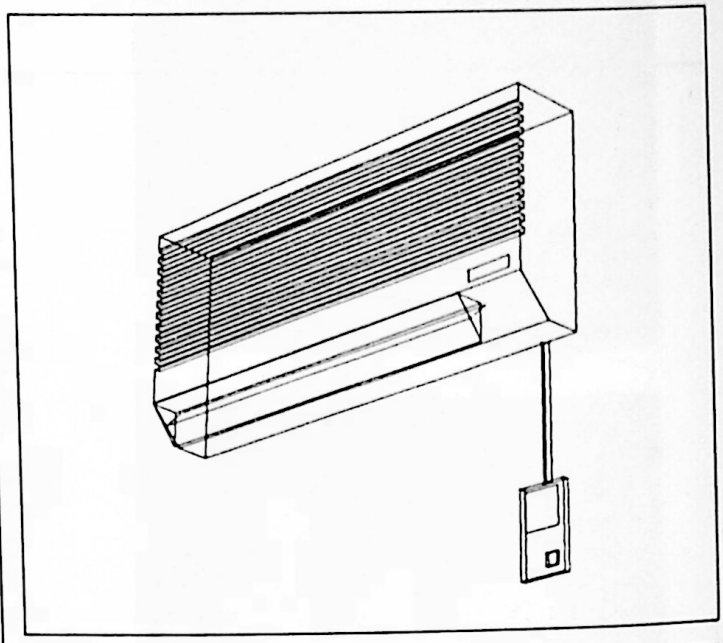


家の間取り

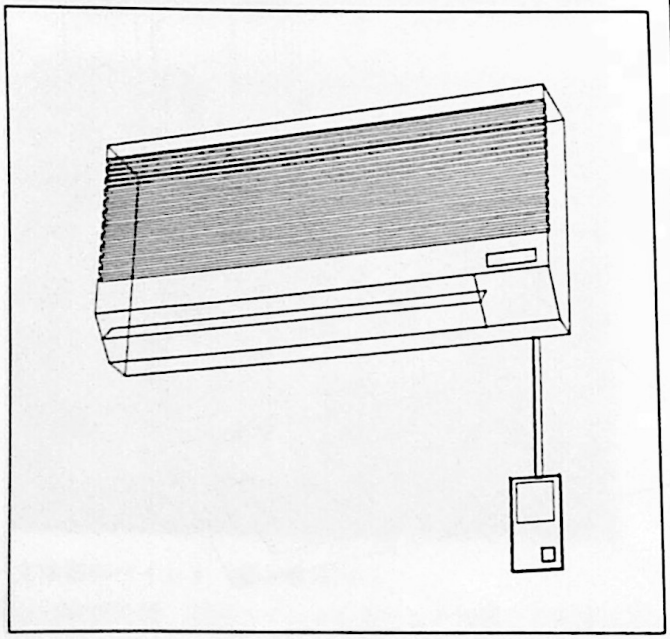


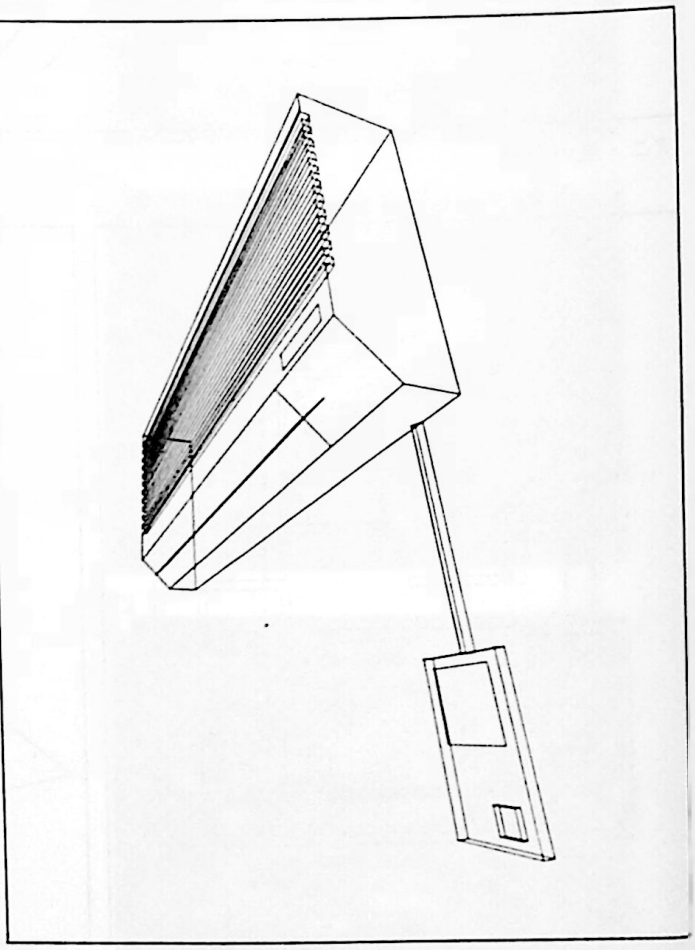
副都心の高層ビル群

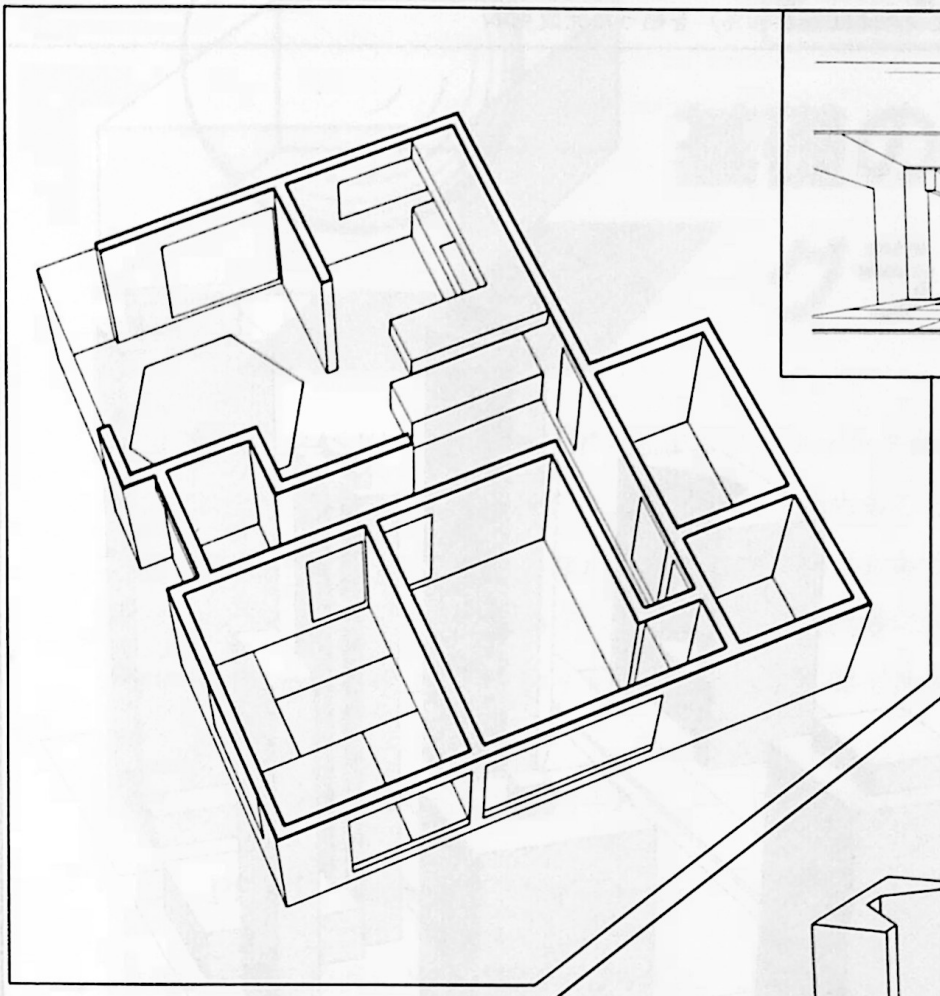


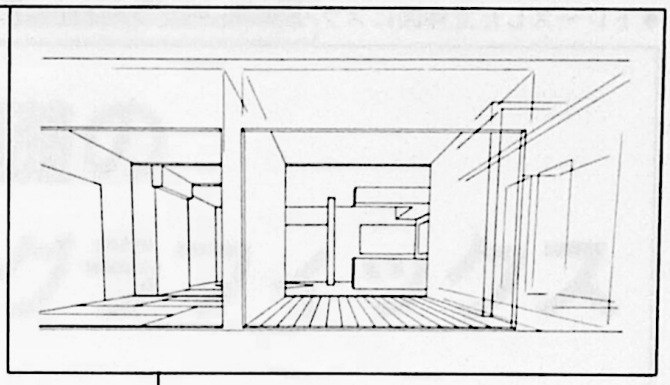


●見る角度を変えてプロッタに出力したエアコンの作画例





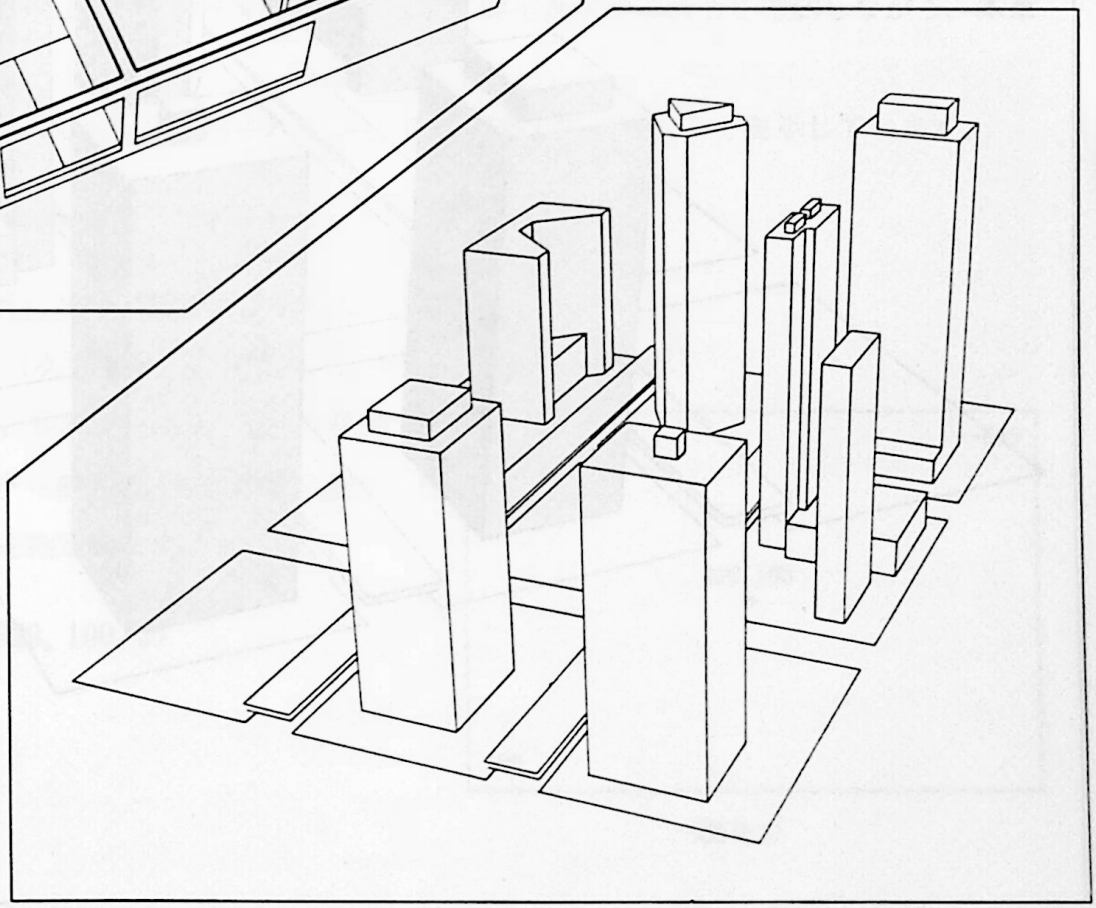




水平位置から見た家の内部

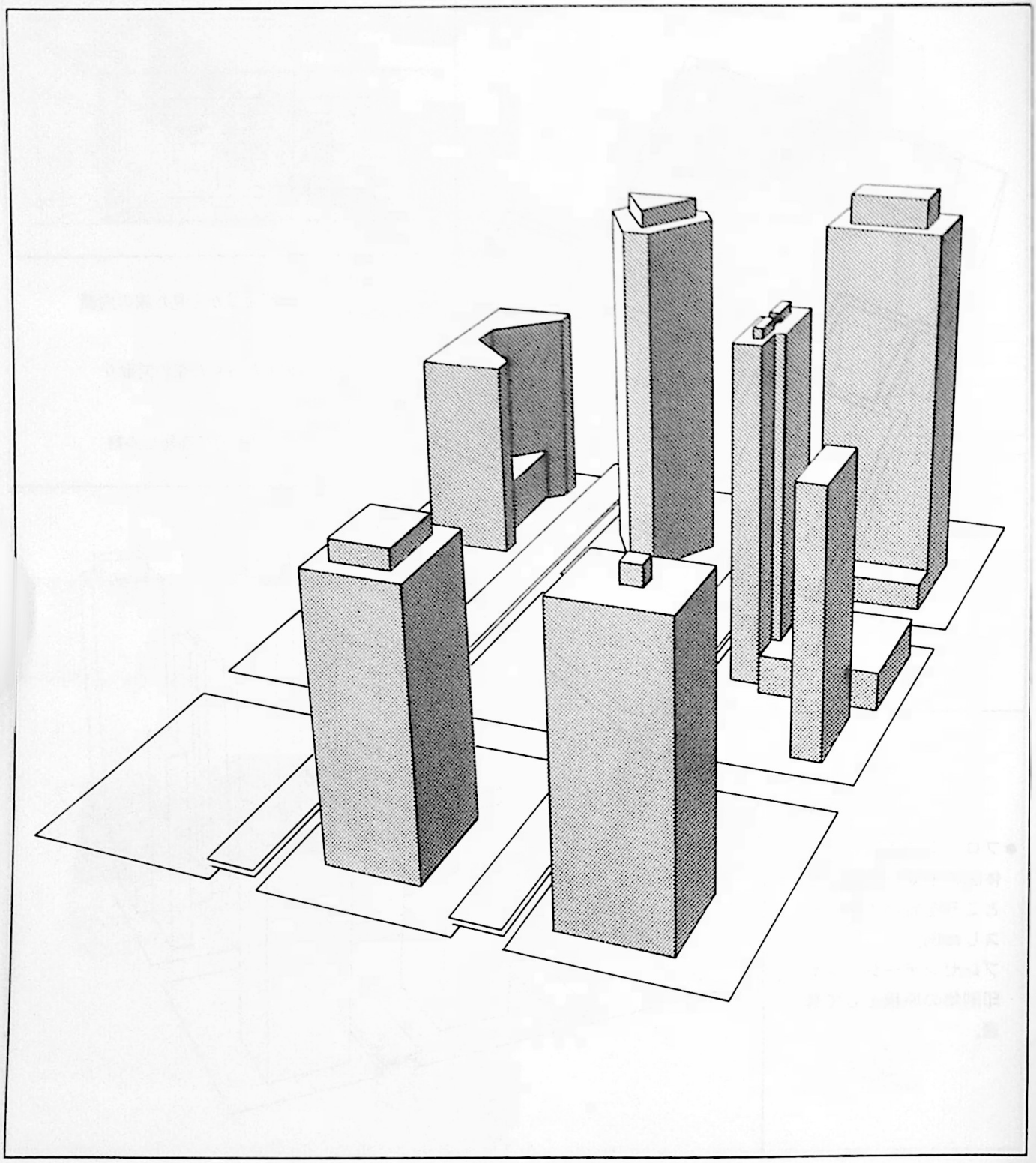
←斜め上から見た家の間取り

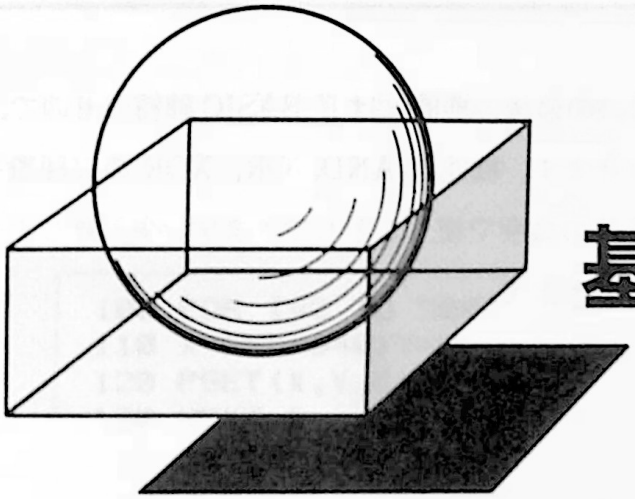
副都心の高層ビル群



● プロッタで作画した立体図の不要（陰線）なところを省いてトレースした例.
プレゼンテーションや，印刷物の原稿として最適.

●トレースした立体図にスクリーントーン（アミ20%＆50%）を貼った完成見本





2章 基礎のグラフィックス

この章はF-BASIC　Ver 3 のサポートするグラフィック命令について解説しています。

ベーシックを学ぶとき，PRINT や INPUT から始めることも意義のあることですが，グラフィックから手掛けると，楽しみながら覚えられる利点もあるようです。グラフィック以外の命令各個については，本体付属のマニュアルや他の入門書に詳しく書かれていますので，本書ではそれらについて逐一述べていません。マニュアル等で基本的な命令を確認しながら，本章を読み進めてください。

なお，本文中の形式の説明で使用される［ ］内は省略してもよいことを示しています。

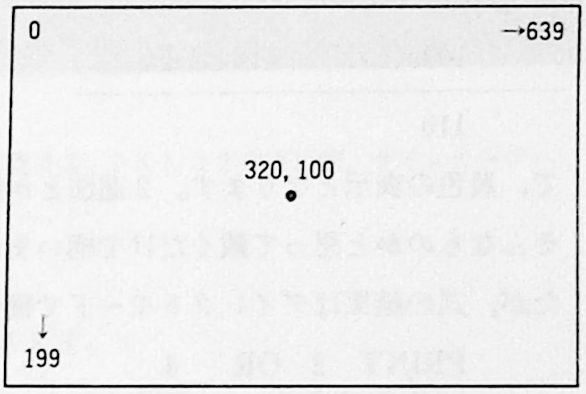
● PSET （point set /ポイントセット）

画面の任意の位置にドットを表示します。

形式　PSET （水平位置，垂直位置［，カラー番号］［，機能］）

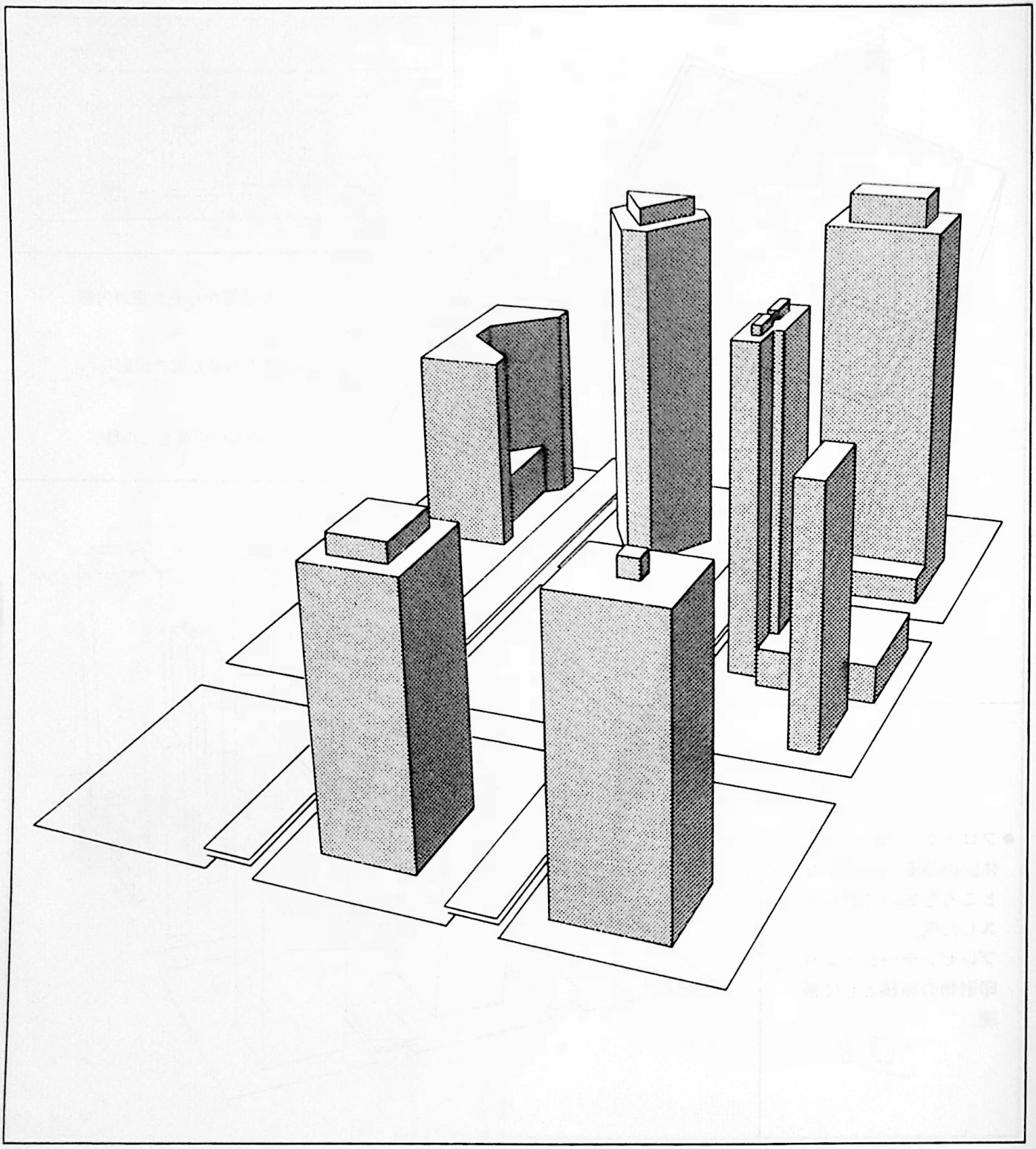
水平位置は0～639，垂直位置は 0 ～199 の範囲で設定します。カラー番号は，その位置に表示させるドットの色を指定します。

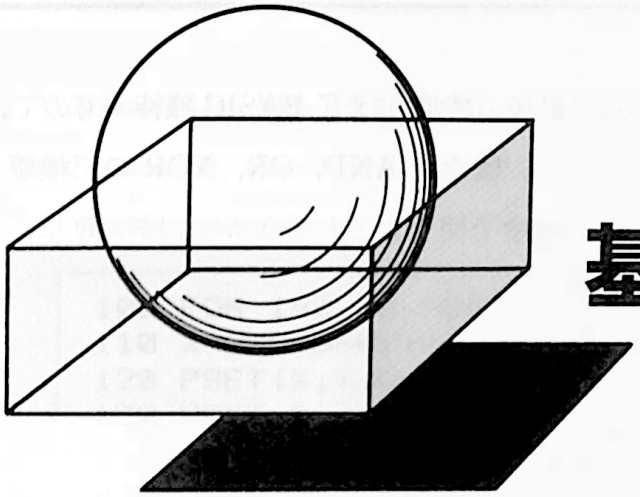
PSET （320, 100, 2）



（図 2-1）

● トレースした立体図にスクリーントーン（アミ20%＆50%）を貼った完成見本





2章 基礎のグラフィックス

この章はF-BASIC　Ver 3 のサポートするグラフィック命令について解説しています。

ベーシックを学ぶとき，PRINT や INPUT から始めることも意義のあることですが，グラフィックから手掛けると，楽しみながら覚えられる利点もあるようです。グラフィック以外の命令各個については，本体付属のマニュアルや他の入門書に詳しく書かれていますので，本書ではそれらについて逐一述べていません。マニュアル等で基本的な命令を確認しながら，本章を読み進めてください。

なお，本文中の形式の説明で使用される［ ］内は省略してもよいことを示しています。

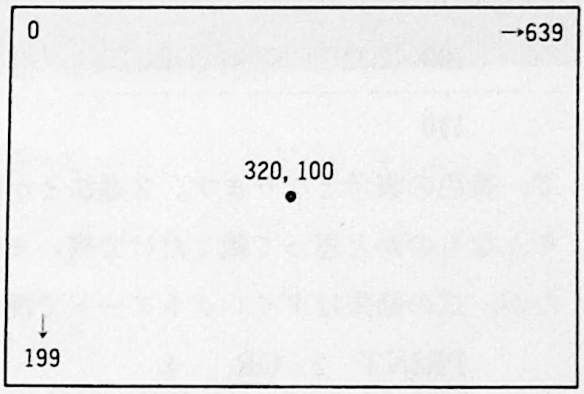
● PSET　(point set /ポイントセット)

画面の任意の位置にドットを表示します。

形式　PSET　(水平位置，垂直位置［，カラー番号］［，機能］)

水平位置は0～639，垂直位置は 0～199 の範囲で設定します。カラー番号は，その位置に表示させるドットの色を指定します。

PSET　(320, 100, 2)



(図 2-1)

これで画面中央に赤い点がセットされます。形式の最後の機能とはF-BASIC独特のもので，背景色と表示色の論理演算の結果の色を表示させます。機能はAND，OR，XORの三種類を指定でき，PSET以外にLINEやCONNECT等の命令で使うこともできます。少し難しくなりますが，参考に論理演算の結果を次に示しておきます。

```
10 PSET (320, 100, 2)
20 PSET (320, 100, 4, ~~~~)
```

~~~~部をORとすると

```
010 (BIN)  ←  2 (DEC)
100 (BIN)  ←  4 (DEC)
-----------------------
110
```

となりOR（論理和）は6になり，黄色のドットが表示されます。なおBINは2進法，DECは10進法を表しています。

AND（論理積）をとると

```
010 (BIN)  ←  2 (DEC)
100 (BIN)  ←  4 (DEC)
-----------------------
000
```

となり，表示色は黒で背景色と区別がつかなくなります。

同様にXOR（排他的論理和）にすると

```
010 (BIN)  ←2 (DEC)
100 (BIN)  ←4 (DEC)
-----------------------
110
```

で，黄色の表示となります。2進法とか論理演算とか耳慣れない言葉がでてきましたが，今はそんなものかと思って戴くだけで構いません。また，2進法の演算式をゴチャゴチャ入れましたが，式の結果はダイレクトモードで簡単に求めることもできます。

```
PRINT 2 OR  4
PRINT 2 AND 4
PRINT 2 XOR 4
```
~~~~

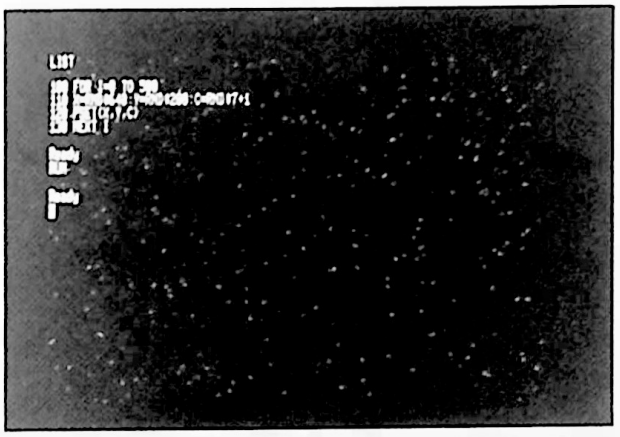
リスト 2-1 は，XとYを乱数によって決め，画面に点を 500 個描かせました。宇宙に浮かぶ星々のようです。リスト 2-2 はサインカーブを PSET で描かせました。

(リスト 2-1)

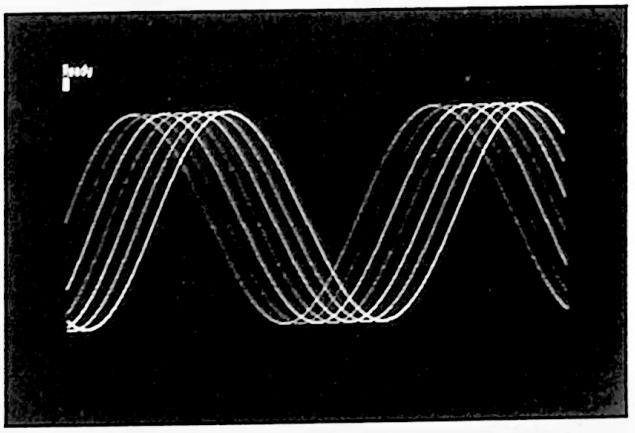
```
100 FOR I=0 TO 500
110 X=RND*640:Y=RND*200:C=RND*7+1
120 PSET(X,Y,C)
130 NEXT I
```

(リスト 2-2)

```
100 CLS
110 FOR I=1 TO 7
120 FOR J=0 TO 639
130 X=SIN((J-20*(I-1))/60)*60
140 PSET(J,100-X,I)
150 NEXT:NEXT
```



(写真 2-1　リスト 2-1 PSET の実行画面)



(写真 2-2　リスト 2-2 の実行例，サインカーブ)

LINE (line／ライン)

キャラクタまたはドットを用いて線や四角形を描きます。

形式 1　LINE[@][(X1, Y1)]−(X2, Y2), 文字列 [, カラー番号] [, Bまたは BF]

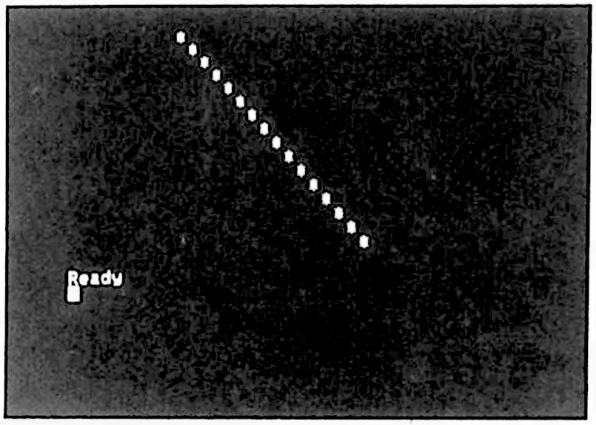
形式 2　LINE[@][(X1, Y1)]−(X2, Y2), 機能 [, カラー番号] [, Bまたは BF]

形式 1 は，文字などのキャラクタを使って線や四角形を表示させます。X1, Y1 などの座標は

WIDTH 文によって規制された範囲をとります。

(リスト2-3)

```
100 WIDTH 40,25
110 LOCATE 0,16
120 LINE(10,0)-(25,15),"*",5
```



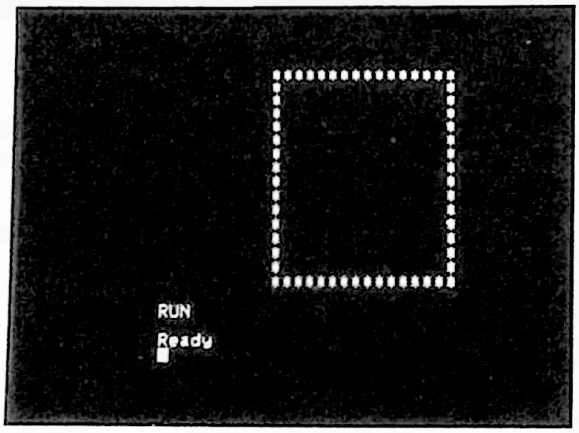
(写真 2-3　リスト 2-3 の実行画面)

ライン文最後のオプションをBとすると，設定した座標の対角線をとる四角形が描画されます。

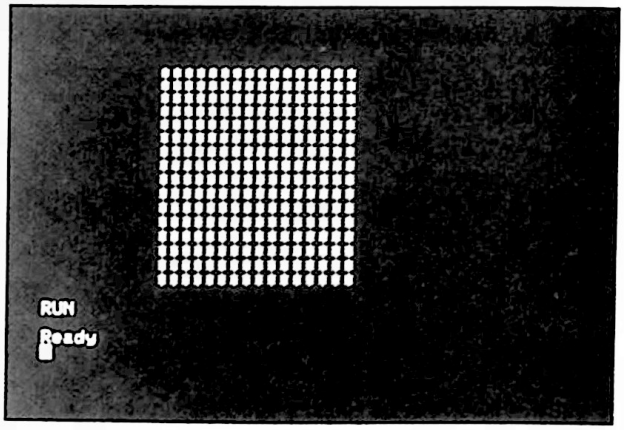
BF にすると,指定したキャラクタで四角形が埋められます。B は BOX, BF は BOX　FILL の略です。

120　LINE(10, 0) − (25, 15), ″*″, 5, B

120　LINE(10, 0) − (25, 15), ″*″, 5, BF



(写真 2-4　B を設定した例)



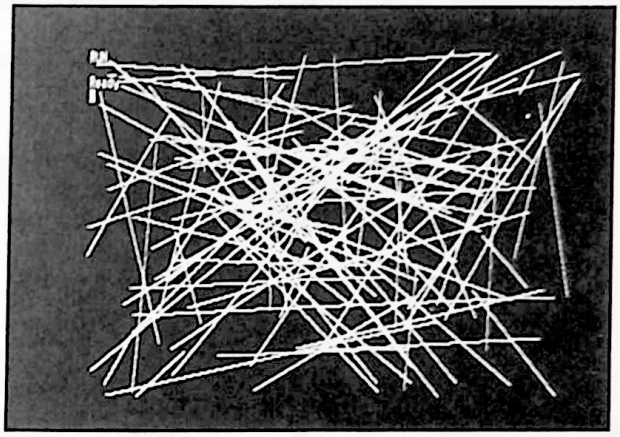
(写真 2-5　BF を設定した例)

形式2は640×200ドットの画面に任意の位置からドットで線を描かせます。最も多く使われる命令のひとつで，グラフィックプログラムには欠かせない命令です。形式は1と似ていますが，文字列のかわりにPSETやPRESETなどの機能を指定します。機能は他にOR, AND, XORが指定でき，これは現在その場所に表示されている色と指定色との論理演算の結果の色を表示します（PSETの項参照）。PSETを指定すると，指定した色がそのまま表示されます。B（BOX），BF（BOX　FILL）のオプションの動作は，形式1と同じく，指定した座標の対角線をとる四角形および，四角形内の塗りつぶしを行ないます。

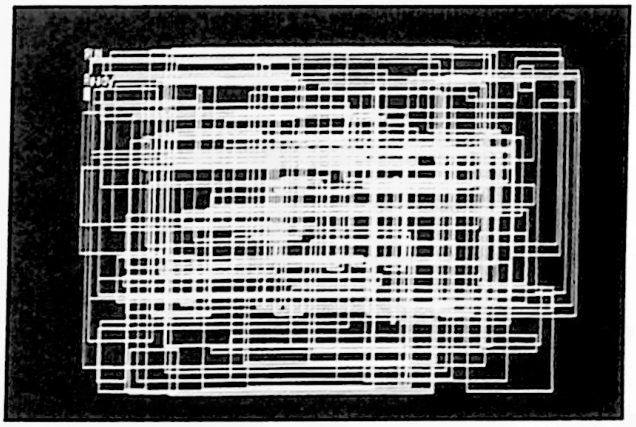
（リスト　2-4）

```
100 FOR I=1 TO 100
110 X1=RND*640:Y1=RND*200
120 X2=RND*640:Y2=RND*200
130 LINE(X1,Y1)-(X2,Y2),PSET,5
140 NEXT I
```

リスト2-4は，XおよびYの座標を乱数で作り，ランダムな線を100回描かせています。



（写真2-6　リスト2-4の実行画面）

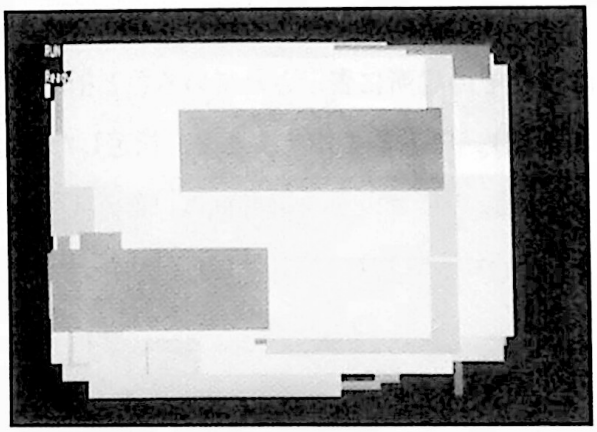


（写真2-7　B(BOX）を指定した場合の画面）

（リスト2-5）

```
100 FOR I=1 TO 100
110 X1=RND*640:Y1=RND*200
120 X2=RND*640:Y2=RND*200
125 C=RND*7+1
130 LINE(X1,Y1)-(X2,Y2),PSET,C,BF
140 NEXT I
```

215行を追加して，指定色も乱数で決め，BF（BOX　FILL）を指定しました。



（写真2-8　リスト2-5でBFを指定した実行画面）

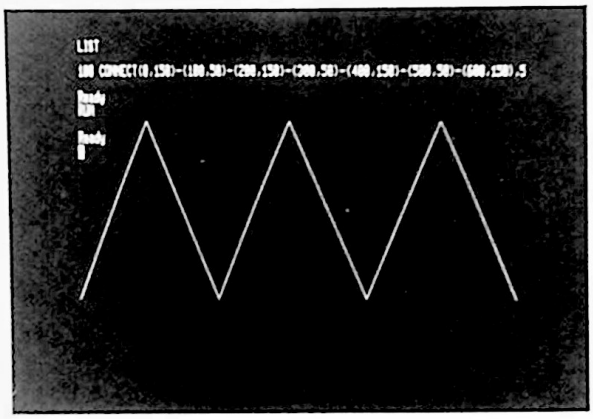
CONNECT　(connect／コネクト)

指定する座標間を直線で結びます。

形式　CONNECT　(X1, Y1)－(X2, Y2) [－(X3, Y3)] … [, [カラー番号] [, 機能]]

LINE命令とよく似ていますが，この命令はいくつもの座標を一度に指定することが可能で，複数の座標間を次々と直線で結ぶことができます。カラー番号と機能の位置はLINE文と反対になりますが，動作は同じです。簡単な図表を描かせる場合などに有効な命令です。

```
100 CONNECT(0,150)－(100,50)－(200,150)－(300,50)－(400,150)－(500,50)－(600,
    150),5
```



（写真2-9　CONNECTの実行）

CIRCLE (circle／サークル)

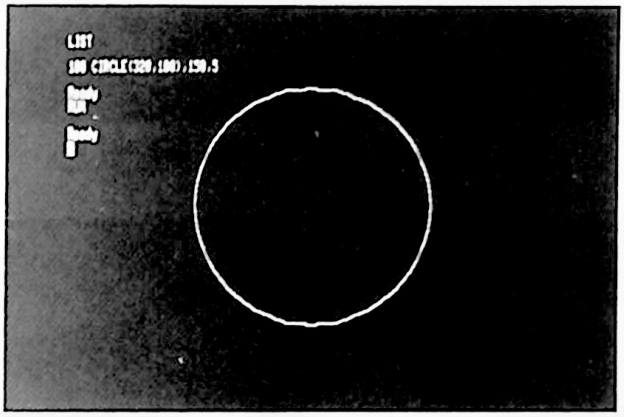
任意の大きさの円や円弧を描きます。

[形式] CIRCLE [@] (円の中心座標X, Y), 半径 [, [カラー番号] [, [比率] [, [開始位置] [, [終了位置] [, [FまたはN] [, 機能]]]]]]

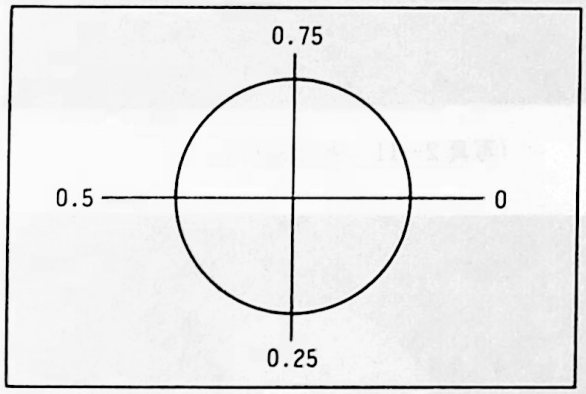
オプションのパラメークがとても多くて混乱しそうですが，それだけに細かい円の指定が可能です。普通の円を描くときは，比率以降を省略して構いません。

```
100  CIRCLE(320,100),150,5
```

これで，画面の中央に半径150で水色の円が描画できます。



(写真 2-10)



(図 2-2)

比率を省略した時は0.4495の値が自動的にとられます。これは640×200ドット画面の水平，垂直比率がこの数値となっているためで，省略した時には真円が描画できます。比率は0～1の範囲で指定でき，楕円の描画も可能です。

開始位置，終了位置では，円の描き始めと終わりの位置を指定できます。範囲は0～1の間で，位置の関係は図2-2の通りです。

開始位置，終了位置の指定を行なうことにより円弧が描画できます。

次のFまたはNは，円あるいは円弧内を塗りつぶすかどうかの指定をします。Fにすると塗りつぶし，Nは何もしません。

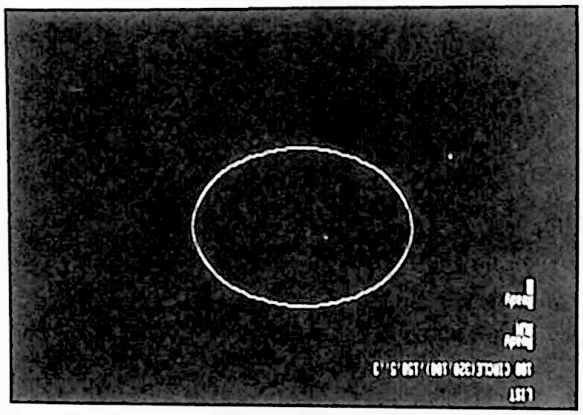
最後の機能は，PSET，PRESET，AND，ORが選べ，省略したときはPSETと見なされます。

① 100　CIRCLE(320,100),150,5,.3　……比率を 0.3 に指定

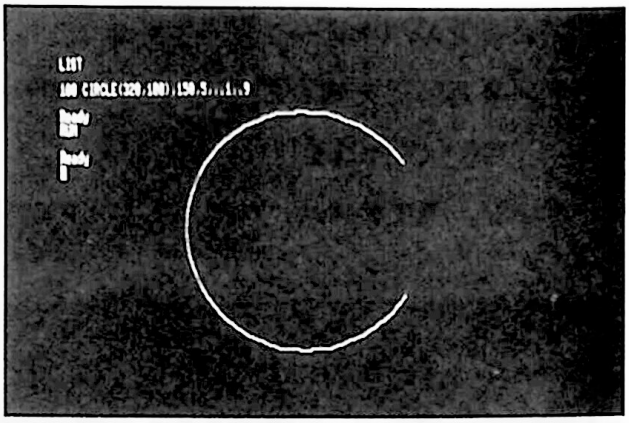
② 100　CIRCLE(320,100),150,5,,.1,.9　……開始，終了位置を指定

③ 100　CIRCLE(320,100),150,5,,.1,.9,F　……塗りつぶしのFを指定

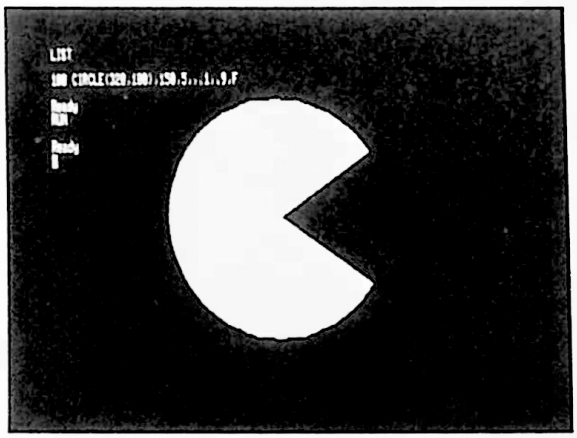
④ 100　CIRCLE(320,100),150,5,,.1,.9,F,XOR……機能の XOR を指定



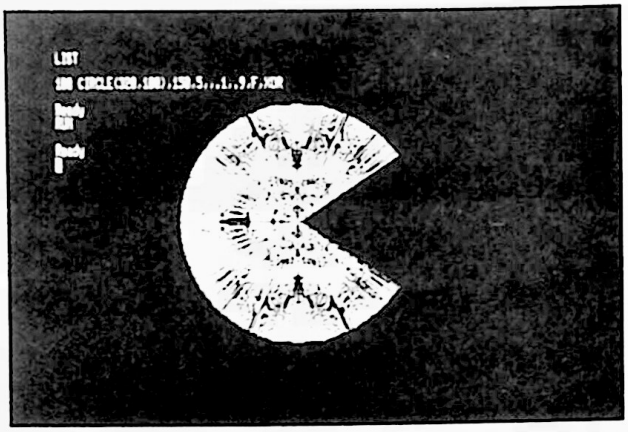
(写真 2—11　実行画面①)



(写真 2—12　実行画面②)



(写真 2—13　実行画面③)



(写真 2—14　実行画面④)

PAINT　(paint／ペイント)

指定する境界色で囲まれた領域を，指定する色で塗りつぶします。

形式　PAINT　(塗り始めの座標 X, Y) [, [カラー番号] [, 境界色 1 [, 境界色 2]…]]

(1) LINE や CONNECT，CIRCLE などによって描かれた図形内を，任意の色で塗りつぶすことができます。

(2)境界色を指定することにより，塗りつぶしの範囲を選択することが可能です。

(3)カラー番号を省略すると，直前の COLOR 文で指定された色で塗られます。

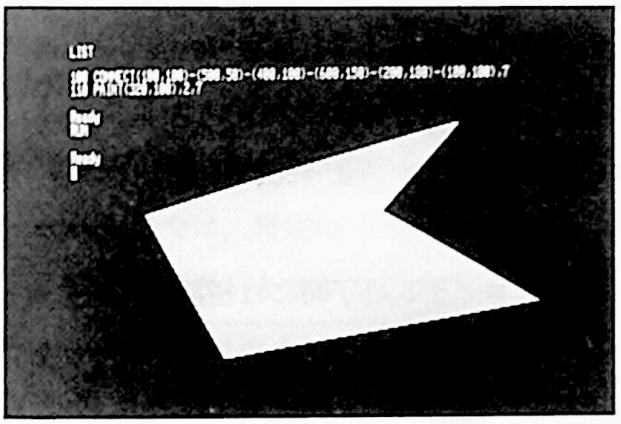
(4)境界色は複数個の指定ができます。また境界色の指定を省略すると，指定したカラー番号が境界色として見なされます。

PAINT を実行するときは，図形を形どっている境界線が，完全に閉じられているかどうかを注意しなければなりません。少しでも開いていると，そこから色がもれだし，みるみるうちにせっかくの図形が塗り消されてしまう場合があります。

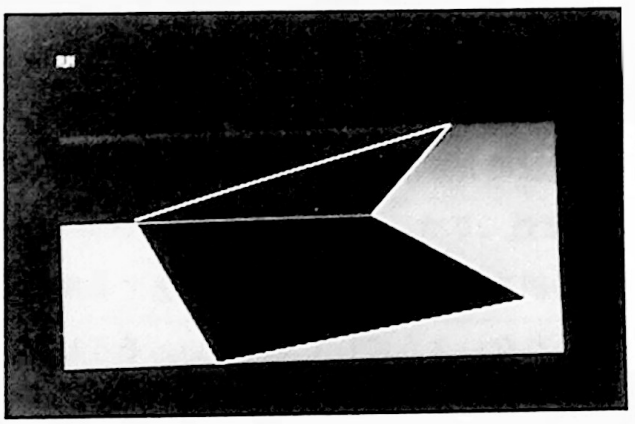
(リスト 2-6)

```
100 CONNECT(100,100)-(500,50)-(400,100)-(600,150)-(200,180)-
    (100,100),7
110 PAINT(320,100),2,7
```

リスト 2-6 は CONNECT 命令で，白の境界線で描かれた図形内を，PAINT 命令で赤に塗りつぶしています。



(写真 2-15　PAINT の実行)



(写真 2-16　色がもれた時の画面)

COLOR　(color／カラー)

画面に表示させる色や背景色を指定します。また，表示されている色を瞬時に変えてしまうパレット機能の動作を行ないます。

形式 1　COLOR　[表示色のカラー番号] [, 背景色のカラー番号]

形式 2　COLOR＝(パレットコード, カラーコード)

形式1は，表示色と背景色の設定を行ないます。

形式2は重要で，表示されている色を瞬時に任意の色に変えることができます。いわゆるパレット機能というもので，動きをだすアニメーションなどはこの機能の応用で実行できます。

（パレットコード，カラーコード）の関係は，

（変更させるカラー番号，表示させるカラー番号）

となります。たとえば

COLOR＝(2, 4)

パレットコード		カラーコード
0	,	0　(黒)
1	,	1　(青)
2	,	2　(赤)
3	,	3　(紫)
4	,	4　(緑)
5	,	5　(水色)
6	,	6　(黄)
7	,	7　(白)

とすると，画面に表示されている赤色（カラー番号2）が緑色（カラー番号4）に変化します。また，この設定以後，赤を表示させる命令を実行しても，緑に表示されます。

パレットの設定は（○色を，△色に）と考えればわかりやすくなります。

電源投入直後のパレットコード，カラーコードの初期設定は次のように，1対1に対応しています。

この組み合わせをどのように変えても構いませんが，プログラム終了時には初期値に戻すことを忘れないようにしてください。そうしないと，次に実行するプログラムやリスト表示が思わぬ色になってしまい，まごついてしまいます。

```
FOR P=0 TO 7 : COLOR = [P,P] : NEXT P
```

←(パレットを初期化するプログラム)

パレット機能の応用例として，PSETの項で示したサインカーブを動かすプログラムを作ってみました（リスト2-7）。各色で描かせたサインカーブを，一目パレットで全色を黒に変えた後，ひとつずつ緑色で表示させました。サインカーブが波打つ様子がわかると思います。プログラムを止める時は，どれかキーを押してください。340行のINKEY$の判断で，押された場合には350行に実行を移動し，パレットの初期化を行なって終了します。

(リスト2-7)

```
100 CLS
190 FOR I=1 TO 7
200 FOR J=0 TO 639 STEP 2
210 X=SIN((J-26*(I-1))/30)*30
220 PSET(J,100-X,I)
230 NEXTJ,I
300 FOR P=1 TO 7:COLOR=(P,0):NEXT
310 FOR P=1 TO 7:COLOR=(P,4)
320 FOR T=0 TO 200:NEXT
330 COLOR=(P,0):NEXT
340 IF INKEY$=""THEN 310
350 FOR P=1 TO 7:COLOR=(P,P):NEXT
```

SCREEN (screen／スクリーン)

R, G, B の各 VRAM の書き込みと表示をコントロールします。

形式 SCREEN [アクティブ VRAM コード] [, ディスプレイ VRAM コード]

アクティブ VRAM コードは書き込める VRAM を示し，ディスプレイ VRAM コードは，表示させる VRAM をさします。

VRAM とは VIDEO RAM の略で，FM-7 は R (レッド)，G (グリーン)，B (ブルー) の3画面分，つまり 640×200 ドット×3画面，計 48 Kバイトの VRAM を持っています。SCREEN 命令は，特定の VRAM を指定して絵を描かせるなどの書き込みをしたり，任意の VRAM だけを表示させることができます。

アクティブ VRAM コード	意味
0	全 VRAM が書き込み不可
1	Bのみ書き込み可
2	Rのみ書き込み可
3	BとRが書き込み可
4	Gのみ書き込み可
5	BとGが書き込み可
6	RとGが書き込み可
7	全 VRAM が書き込み可

ディスプレイ VRAM コード	意味
0	どの VRAM も表示しない
1	Bのみ表示
2	Rのみ表示
3	BとRを合成して表示
4	Gのみ表示
5	BとGを合成して表示
6	RとGを合成して表示
7	全 VRAM を表示

アクティブおよびディスプレイ VRAM コードは，0から7の範囲をとり，内容は以下の通りです。

(例) ①　SCREEN　1,2　←(青の VRAM が書き込み可能で，赤の VRAM だけを表示)

②　SCREEN　4：CLS　←(緑の VRAM の内容が消去される)

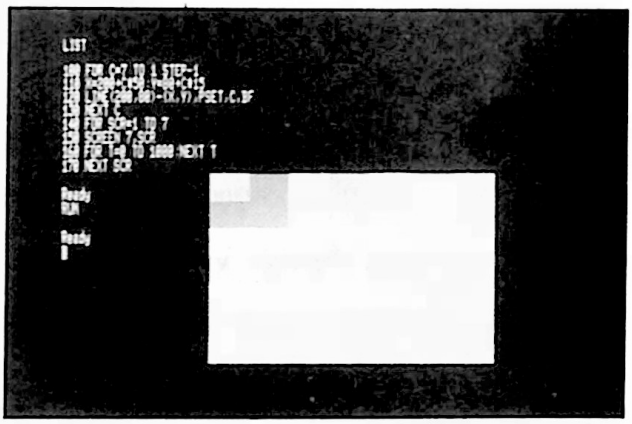
パレットと似ている感じで混同しそうですが，動作は違いますので注意が必要です。SCREEN 文を操作しているとき，カーソルも含めて何も表示されなくなった場合は[PF9]あるいは「SCREEN　7, 7」を打ち込み初期化してください。

```
100 FOR C=7 TO 1 STEP-1
110 X=200+C*50:Y=80+C*15
120 LINE(200,80)-(X,Y),PSET,C,BF
130 NEXT C
140 FOR SCR=1 TO 7
150 SCREEN 7,SCR
160 FOR T=0 TO 1000:NEXT T
170 NEXT SCR
```

(リスト 2-8)

150 行で，ディスプレイ VRAM コードの値を変えて，各色の表示の様子を示しました。

160 行は，表示時間のタイミングをとっているだけです。



(写真 2-17　SCREEN の実験)

SCREEN 文とパレットを上手に使うと，R, G, B 3 画面のうちのひとつを表示しながら別の画面に作画して，作画終了と同時に画面を切り替える芸当ができます。この動作を連続して繰り返えすとアニメーションが可能です。しかし，本物のアニメーションのような，滑らかな動きを出すことは作画スピードの点で難しく，コマ送り風の動画になってしまいます。

SYMBOL　(symbol／シンボル)

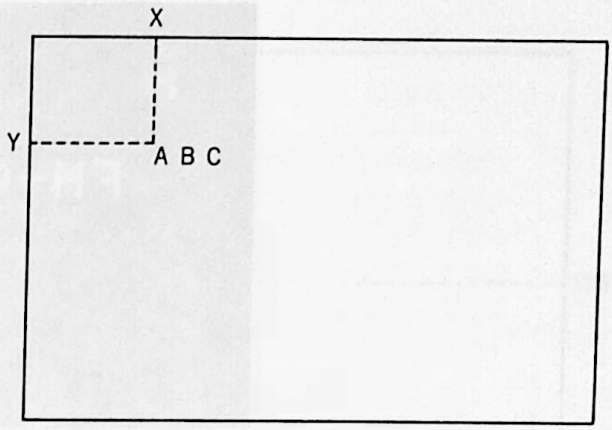
文字列を任意の大きさ，任意の位置，任意の方向で表示します。

[形式]　SYMBOL　(X, Y)，文字列，横倍率，縦倍率 [，[カラー番号] [，[角度コード] [，機能]]]

この命令はF-BASIC独特のもので，文字列を好みの大きさに拡大でき，画面のタイトル表示によく使われます。また文字列は90°単位で回転させることもできます。

（X，Y）は表示位置を示し，文字列の枠内の左上のドット座標を指します。

倍率は，拡大の倍数を表し，横方向，縦方向別々に指定することが可能です。



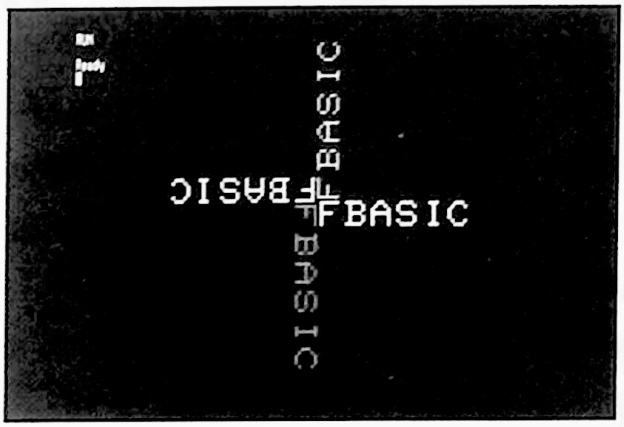
(図 2-3)

角度コードは文字列の回転方向を示し，90°ずつ左回転させて表示できます。

機能は，PSET，PRESET，AND，OR，XOR，NOT の指定ができます。

(リスト 2-9)

```
100 SYMBOL(320,100),"FBASIC",4,2,5,0
110 SYMBOL(320,100),"FBASIC",2,4,4,1
120 SYMBOL(320,100),"FBASIC",4,2,3,2
130 SYMBOL(320,100),"FBASIC",2,4,2,3
```

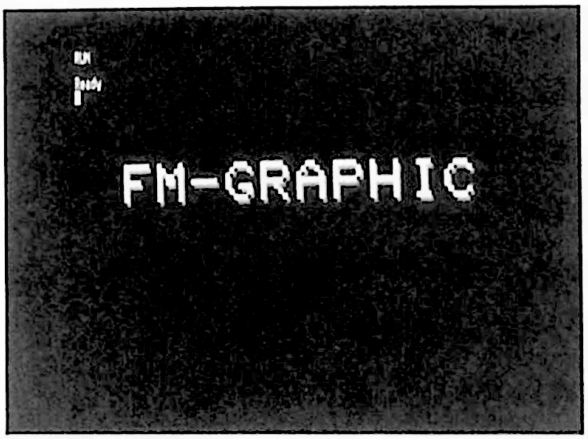


(写真 2-18 リスト 2-9 の実行画面（文字列の回転))

SYMBOL 命令を使って，影付き文字を表現してみました。表示位置をわずかにずらせて，文字列の二重描きをさせています。小イキなタイトル表示として使えます。

(リスト 2-10)

```
100 SYMBOL(70,70),"FM-GRAPHIC",6,3,2
110 SYMBOL(66,68),"FM-GRAPHIC",6,3,5
```



(写真 2-19　リスト 2-10 の実行画面（影付き文字））

GET@　(get@／ゲットアット)

指定した範囲の画面情報を配列に取り込みます。

形式 1　GET@(X1, Y1)－(X2, Y2)，配列名

形式 2　GET@A(X1, Y1)－(X2, Y2)，配列名

形式 3　GET@(X1, Y1)－(X2, Y2)，配列名，G［，カラー番号［，カラー番号］…］

形式 4　GET@A(X1, Y1)－(X2, Y2)，配列名，G

ひとつのキーワードに形式が 4 つもあるとうんざりしてきます。しかし，それだけに，きめ細かいコントロールができると思ってしっかり理解してください。

GET@は，後述の PUT@と対で使われるもので，指定する場所の画面情報をそっくり配列の中に取り込んでしまう機能を持ちます。一度配列に入れてしまうと，PUT@で呼びだせば即座にその情報を画面の任意の場所に表示することができます。

形式 1 および 2 は，画面情報としてキャラクタを，形式 3 および 4 はドットを取り込みます。また形式 2 および 4 は，カラー情報も含めて配列に格納します。

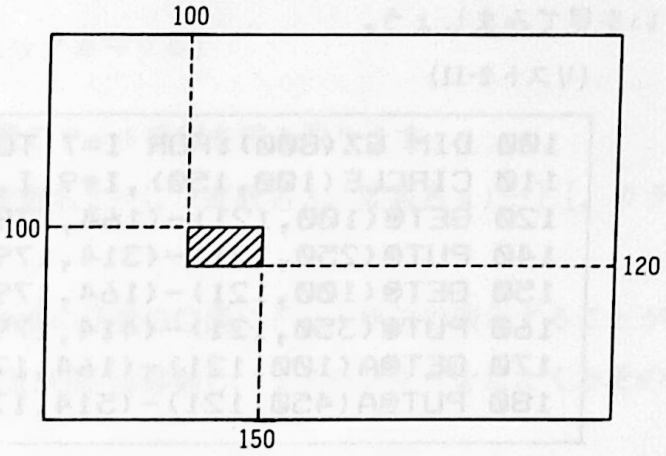
形式＼対象	キャラクタ単位	ドット単位	カラー情報
形式　1	○	×	×
形式　2	○	×	○
形式　3	×	○	×
形式　4	×	○	○

GET@で配列に取り込む要素

グラフィックスを行なううえで，重要な形式は3および4に絞られますので，本書では，これらを中心に述べていきます。

(X1, Y1)－(X2, Y2) は，配列に入れる画面範囲を示します。たとえば，

(100, 100)－(150, 120) は図2-4の範囲をとります。



(図2-4 (100,100)－(150,120) の範囲)

説明が前後しますが，GET@文を実行するときは，あらかじめ取り込むべき画面範囲に従った大きさの配列を宣言しておかねばなりません。

1ドットが1ビットに対応していますので，1バイトのメモリーで8ドット分が格納できます。画面範囲が1000ドットある場合は，

1000ドット÷8ビット＝125バイト

となり，125バイト分のメモリーが必要です。これを配列変数に置き換えると，整数型は1つの変数に2バイト分使うので

125バイト÷2＝62.5

となり，63個の配列を宣言しておけば良いわけです。

単精度型では　　125バイト÷4＝31.25

倍精度型では　　125バイト÷8＝15.625

となり，それぞれ32個，16個の配列を用意します。

富士通のマニュアルでは整数型の配列に格納する例を挙げていますが，要は必要なメモリーの数が確保できれば良いわけで，どの型の配列変数を使っても構いません。

形式3に必要な配列の個数＝INT((INT((総ドット数＋7)／8)＋a－1)／a)

aの値は　整数型 … a＝2，単精度型 … a＝4，倍精度型 … a＝8

形式4に必要な配列の個数＝INT((INT((総ドット数＋7)／8)＊3＋a－1)／a)

形式4のカラー情報も含めたときの配列の大きさは形式3の場合の約3倍になります。

形式3のカラー番号は，取り込むべきドットの色を指定します。指定された色で表示されているドットだけを読み取り，配列に格納されます。各形式における画面情報の取り込み方の違

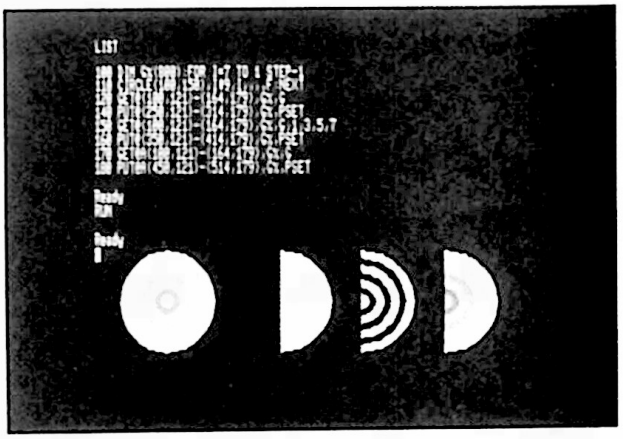
いを見てみましょう。

(リスト2-11)

```
100 DIM G%(800):FOR I=7 TO 1 STEP-1
110 CIRCLE(100,150),I*9,I,,,,F:NEXT
120 GET@(100,121)-(164,179),G%,G
140 PUT@(250,121)-(314,179),G%,PSET
150 GET@(100,121)-(164,179),G%,G,1,3,5,7
160 PUT@(350,121)-(414,179),G%,PSET
170 GET@A(100,121)-(164,179),G%,G
180 PUT@A(450,121)-(514,179),G%,PSET
```

写真左端の同心円状の図が，110行で描かれた原図です。その隣りの半円状のものは，GET @の書式の違いで，配列に取り込まれた画面情報がどのようになっているのかをPUT@命令を使って表わしています。

左から，形式3でカラー番号を省略した場合(120行)，カラー番号を指定した場合(150行)，形式4の場合（170行）の順で並んでいます。



(写真2-20　リスト2-11の実行画面（画面情報の取り込み方))

なお，GET@で取り込む画面の範囲は，原図の右半分だけを指定しました。

PUT@　(put@／プットアット)

GET@で配列に格納した画面情報を任意の位置に表示します。

形式1　PUT@(X1,Y1)－(X2,Y2)，配列名［，カラー番号］

形式2　PUT@A(X1,Y1)－(X2,Y2)，配列名

形式3　PUT@(X1,Y1)－(X2,Y2)，配列名，機能［，カラー番号］

形式4　PUT@A(X1,Y1)－(X2,Y2)，配列名，機能

各形式は，GET@の形式と対応しています。形式1，2はキャラクタ単位，3，4はドット単位で(X1,Y1)－(X2,Y2)の画面範囲に表示します。GET@で宣言した配列の大きさを超える画面範囲の指定はできません。

GCURSOR (gcursor／グラフィックカーソル)

画面に表示したグラフィック・カーソル位置のドット座標を読み取ります。

形式 GCURSOR(X, Y), (変数名1, 変数名2) [, (変数名3, 変数名4)…] [, カラー番号]

GCURSOR は, ＋の形をしたカーソルで, 画面の任意の位置にドット単位で表示することができます。表示後カーソルキーを押すと GCURSOR は移動し, リターンキーをたたくとその位置のドット座標を変数に代入します。

GCURSOR (320, 100), (A, B)

これを打ち込むと, 画面中央に GCURSOR が表示されます。次に, カーソルキーを使って GCURSOR を任意の位置へ移動させ, リターンキーを押すと, その位置の水平座標が変数Aに, 垂直座標がBに代入されます。念のため, ダイレクトモードで PRINT A, B と打ち込んでください。読み込まれた座標値が確認できます。

変数名は複数個の指定が可能で, カーソルキー, リターンキーの操作の順で次々と座標を読み取ることができます。GCURSOR はシフトキーを併用すると移動速度が速くなります。

カラー番号は, GCURSOR の表示色を指定します。

GCURSOR は他の機種にはみられない F—BASIC 独特のもので, ライトペンがわりとして使える便利な命令です。ライトペンは精度に問題が残りますが, 操作性にすぐれています。GCURSOR は反対に, 精度はドット単位で正確なのですが, キー操作がわずらわしいといえます。一概にどちらが良いかとはいえませんが, 初期投資をしなくて済む分だけ GCURSOR に軍配が上がります。マウスは, この両方の特長を兼ね備えたものといえます。富士通から FM用のマウスも発売されています。

CONSOLE (console／コンソール) 省略形 CONS.

スクロール範囲や, ファンクションキーの表示スイッチの設定を行ないます。

形式 CONSOLE [スクロール開始行] [, [スクロール行数] [, [ファンクション表示スイッチ] [, コンソールカラースイッチ]]]

スクロール開始行は0～24，行数は0～25の範囲で指定します。この指定は WIDTH で設定された制限を受けます。

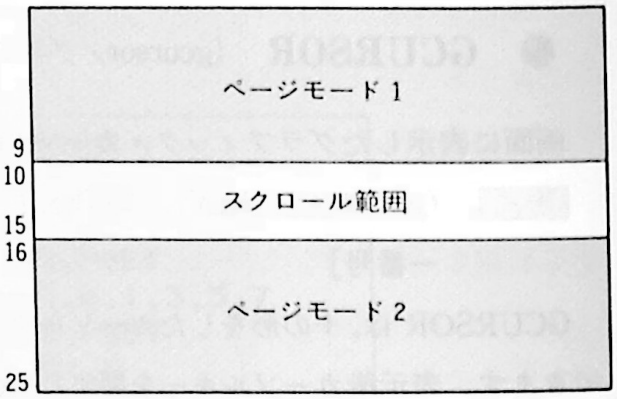
開始行および行数の指定により，画面を分割させることができます。「WIDTH　80,25」のとき，「CONSOLE　10,5」は，スクロール範囲が10行目から15行目までとなり，9行目までがページモード1，16行～25行目がページモード2に設定されます。

市販のアドベンチャーゲームなどでは，CONSOLE　22,3と初期設定されていて，ページモード1に絵を，下3行のスクロール範囲に状況説明を表示させている例が多いようです。

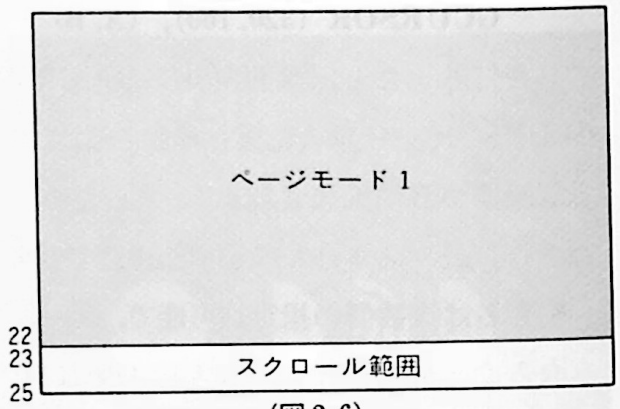
ファンクションキー表示スイッチを1にすると，画面の下2行にファンクションキーの内容が表示されます。この場合，スクロール開始行および行数の範囲は2行分少なくなります。

コンソールカラースイッチを1にすると，画面は緑色だけの単色表示になり，文字の出力動作が速くなります。

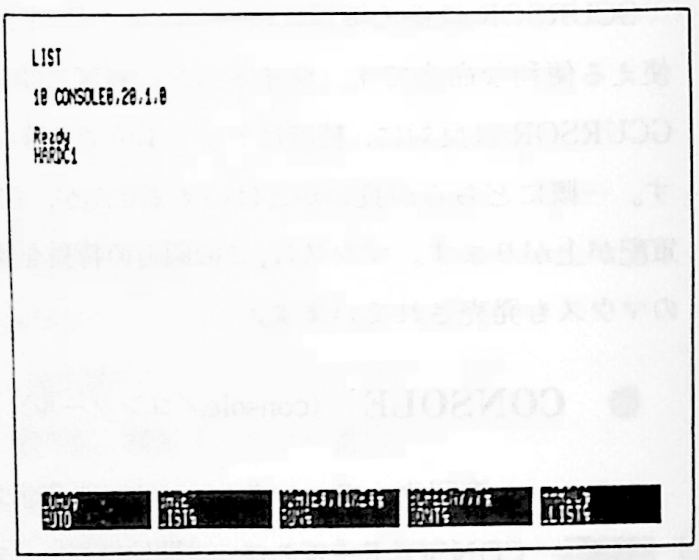
電源投入時は，「CONSOLE 0,20,0,0」に初期設定されています。



(図2-5)



(図2-6)



(図2-7　ファンクション表示スイッチオン)

CLS (clear screen／クリアスクリーン)

画面を消去し，カーソルをホームポジションに戻します。

形式　CLS（消去範囲コード）

なにをいまさらと思われるかも知れませんが，CLS の消去範囲コードというパラメータに気が付く人は，以外と少ないものです。

消去範囲は，CONSOLE 命令の設定と密接な関係を持っています。

消去範囲コード	意　味
0	全画面をクリア
1	スクロール画面をクリア
2	ページモード１をクリア
3	ページモード２をクリア

図 2-8

◆F-BASIC 活用ソフト　　（山海堂発行）

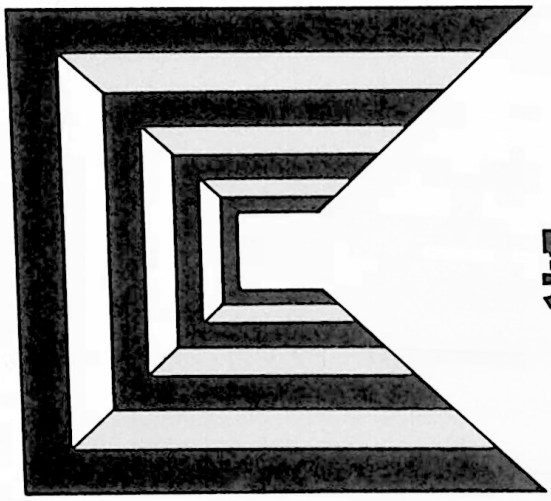
ワープロとグラフ　(FM-7・NEW7・77)

潮　　守著／Ｂ5判／2,100円

FM-7・NEW7・77で使えるプログラムのリストとその使い方・操作法を収載.
●辞書付簡易日本語ワープロ．縦印刷もできます。
●ビジネスグラフと統計表・統計グラフ．複合グラフも表示できます.
●ディスクユーティリティ．アクセスのスピードをアップするフォーマットなど８種.
●ＦＭ⟷ＰＣファイル変換．ＦＭとＰＣ間で相互にプログラムを読み込めます.
●そのほか，グラフィックパネル・金利計算・オルガンなど計11本収載.

フロッピーディスク頒布サービス

全プログラム17本をディスクで頒布しております．5インチ，3.5インチとも7000円です(送料当社負担)．サイズを明記のうえ現金書留にてお申し込み下さい.
申し込み先　　東京都文京区本郷　5-5-18　山海堂　営業部　ＭＳ係



3章 実践プログラミング

プログラミングのカンどころ

グラフィックの命令について，その機能や動作など一通り述べてきましたが，ある目的を達成するためのまとまったプログラムを組もうとすると，ある程度の慣れやカンが必要となってきます。

また，マイコン教室に通ってベーシックを習ったけれど，いざ自分で具体的なプログラムを作ろうとしても，どうしてよいやら困ってしまうという話もよく耳にします。

プログラミングは，キーワードをいくら正確に覚えようとも，英単語だけを覚えて英会話をしようとするのと同様，ぎこちないものとなるばかりでなく，まったく意味の通らない場合さえ起こります。プログラムは〝流れ〟ですから，目的を達しつつ流れが止まらないように道を作っていかねばなりません。キーワードの機能や動作を理解することはもちろん重要なことですが，もっと大切なことは，各キーワードをどのように組み合わせ，いかにしてプログラムとしてまとめるかという点です。そのためには，自分のさせたい仕事をなるべく小さな単位に分け，単位ごとのプログラムを作って，最終的にそれらをつなぎ合わせて一本のプログラムとして仕上げる方法をとることが最良ではないかと思います。

フローチャートを書いてみることも一手段でそれなりの効果は絶大ですが，これは以外と大変なことで，ベーシックを覚えたての時にはとまどうばかりで終わってしまいがちです。

ベーシックは，制約の少ない気軽に書き下せる言語です。させたい仕事を細かく分けてプログラムを組み，その小さなプログラムをサブルーチンとして，メインプログラムから呼びだす

ようにすれば，自然と大きなプログラムに育っていきます。まずは，単一機能の小さなプログラムを気軽に書き下すことから始めてください。

よくIF文を使い過ぎるなとか，GOTO文をところ構わず使用することは良くないとかいろいろいわれていますが，最初のうちはそんなことにとらわれる必要はありません。どんな命令の使い方をしようと，エラーを起こさずに目的を達することができれば，それで立派なプログラムといえます。プロになるのであるならば話は別ですが，最初から，プログラミングは，ああすべきだ，こうあるべきだと窮屈に考え過ぎてしまっては，自由な発想は生まれません。

この命令を使えば，こんなことができるのではないかとか，あるいは，こうさせたい場合はどの命令を使えば良いのかなどと，いろいろ考え，あれこれ試してみることが上達への第一歩です。

この章では，具体的なプログラミング例として，グラフィックを中心にいくつかのサンプルプログラムを掲載しました。各命令の使い方やテクニックなどを自分なりにくみ取りプログラム作りに役立ててください。

グラフィック・デモンストレーション

手始めに，グラフィック命令を使った簡単なデモプログラムを作ってみましょう。デモはパソコンを手にすると，誰もが一度は作るプログラムのようですが，各命令の機能や動作を理解し，確かめるうえではうってつけといえます。

リスト3-1は，PSET，LINE，CIRCLE，PAINTの命令を使用したプログラムです。座標の値は乱数で決めており，またPAINTのルーチンはサブにしています。

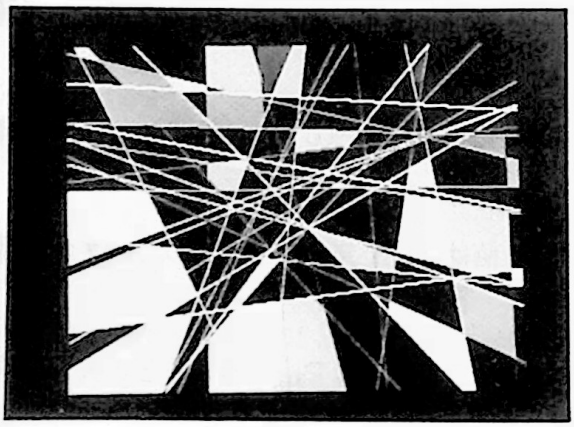
リスト3-2も同様のデモプログラムですが，座標の値はFOR文によって決めています。乱数と異なり，規則性をもたせた図が描画できます。

比較的簡単なプログラムで，意外と面白い画面作りができますので，いろいろと工夫して自分独自のデモプログラム作りに挑戦してください。

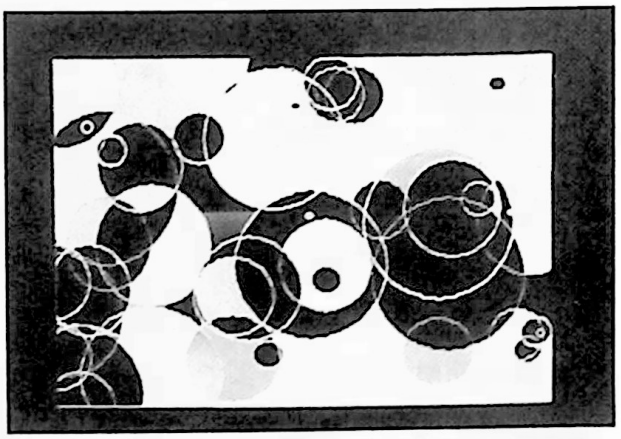
リスト3-3は，表示色を瞬時に別の色に変えてしまうパレットのサブルーチンです。デモプログラムの中で「GOSUB　300」として呼びだすなど，いろいろ活用してください。330行は，パレットコードを初期化するためのルーチンです。

(リスト3-1)

```
100 '*************** PSET ***************
110 RANDOMIZE TIME/3:CLS
120 FOR I=0 TO 100
130 X=RND*640:Y=RND*199
140 PSET(X,Y,RND*7):NEXT
150 '************** LINE ***************
160 FOR I=0 TO 15
170 X1=RND*640:X2=RND*640
180 Y1=RND*200:Y2=RND*200
190 LINE(X1,0)-(X2,199),PSET,RND*6+1
200 LINE(0,Y1)-(639,Y2),PSET,RND*6+1
210 NEXT:GOSUB 270
220 '************* CIRCLE **************
230 CLS:FOR I=0 TO 50
240 X=RND*640:Y=RND*199
250 CIRCLE(X,Y),RND*100,RND*6+1
260 NEXT:GOSUB 270:GOTO 100
270 '************ PAINT/SUB *************
280 FOR I=0 TO 100
290 X=RND*640:Y=RND*199
300 PAINT(X,Y),RND*6+1,1,2,3,4,5,6,7
310 NEXT:RETURN
```



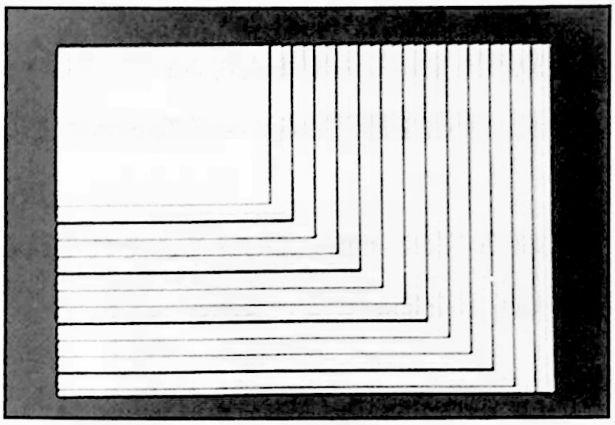
(写真3-1　乱数を使用したデモ画面①)



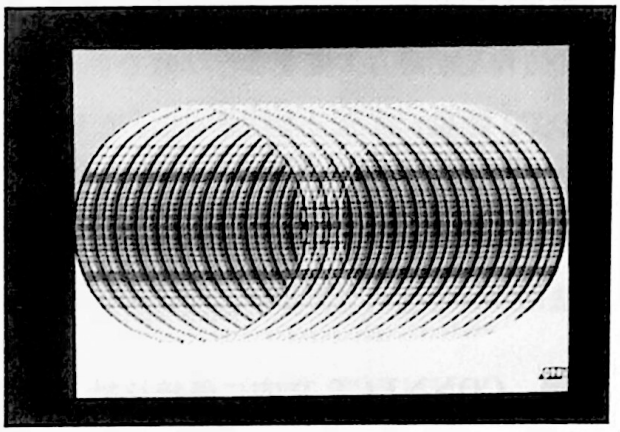
(写真3-2　乱数を使用したデモ画面②)

(リスト 3-2)

```
100 '*************** PSET ***************
110 CLS:FOR X=0 TO 640 STEP 10
120 FOR Y=0 TO 190 STEP 10
130 PSET(X,Y,X MOD 7):NEXT Y,X
140 '*************** LINE ***************
150 FOR I=640 TO 0 STEP-4
160 LINE(0,0)-(I,I/3),PSET,I MOD 7,BF
170 NEXT
180 '************** CIRCLE **************
190 CLS:FOR I=150 TO 494 STEP 4
200 CIRCLE(I,100),150,I MOD 7
210 NEXT
220 '***************PAINT ***************
230 PAINT(10,10),6,1,2,3,4,5,6,7
240 PAINT(10,190),2,1,2,3,4,5,6,7
```



(写真 3-3　規則性を持たせたデモ画面①)



(写真 3-4　規則性を持たせたデモ画面②)

(リスト 3-3)

```
300 '+++++++++++++++ PALET ++++++++++++++
310 FOR I=1 TO 7:COLOR=(I,8-I)
320 FOR T=0 TO 500:NEXT T,I
330 FOR P=0 TO 7:COLOR=(P,P):NEXT
```

パソコンで絵を描こう

パソコンに絵を描かせる方法には，大きく分けて二通りの手段があります。

ひとつは，決められた手順に従って座標のデータを読み取り，LINE 文などで線をつなぎ，絵を一気に描き上げてしまう方法です。パソコン・ショップでアニメキャラクタを描かせているデモンストレーションは，この方法が多いようです。

もうひとつの手段は，画面を見ながらキーボードを絵筆がわりに，直接ディスプレイに描いていく方法です。

■ 絵を描くプログラム

最初に前者の方法について述べてみましょう。まず，下絵を準備し，座標を拾います。座標は絵を形どる要所要所の点を順々にプロットします。点は 640×200 ドットの範囲でなければなりません。座標を拾い終われば作業はおしまいで，あとはこのデータに従って座標を打ち込み，線を結ばせれば絵は完成します。

絵を描かせるプログラム自体はとても簡単ですが，実は，下絵を描いたり座標を読み取ったりする作業が最も大変なことなのです。また下絵の出来不出来がそのまま表現されてしまいますので，下絵には充分な注意を払わなければなりません。下絵を修正しながら座標の入力をしようなどという考えは禁物です。

絵の座標データを格納する方法はいくつかありますので，次に実例をまじえて，それぞれの方法について述べてみます。プログラム例は，FM という形を描かせています。

● CONNECT 文内に直接格納

一番簡単で描画スピードも速いのですが，複雑で混み入った図柄の座標値をズラズラ入れてしまうと，プログラムが見づらくなるとともに，修正がやりにくくなります。比較的単純な図表や絵を描かせるときには有効な手段です。

(リスト 3-4)

```
100 CONNECT(130,50)-(270,50)-(270,70)-(170,70)-(170,80)-(250
    ,80)-(250,100)-(170,100)-(170,130)-(130,130)-(130,50),7
110 CONNECT(320,50)-(360,50)-(410,95)-(460,50)-(500,50)-(500
    ,130)-(460,130)-(460,85)-(410,130)-(360,85)-(360,130)-(32
    0,130)-(320,50),7
120 PAINT(150,60),2,7:PAINT(340,60),1,7
```

● DATA 文に格納し，READ 命令で読み出す

絵を描かせるプログラムとして代表的な方法です。DATA 文を増やすことにより，どんな複雑な図形でも表現可能です。

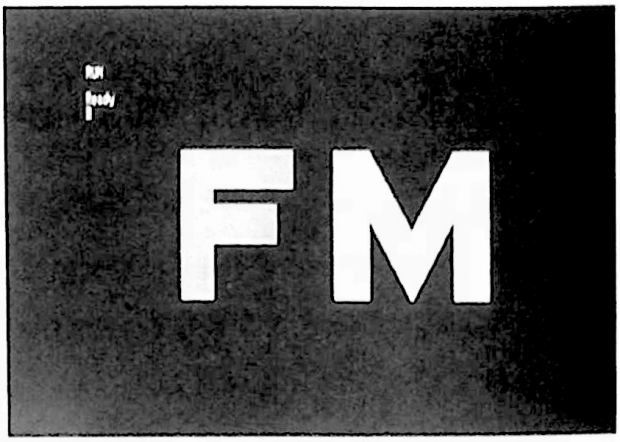
200 行以降のデータを 100 行と 110 行の READ　X，READ　Y で読み取り，LINE 文で線をつなげて絵を描かせています。

描き始めを区別するために，データには少し細工を加えています。通常 LINE 文は，線を結ばせる 2 点の X，Y 座標をセットしますが，次々と線をつなげるときにはデータの重復が問題になります。そこで，一筆描きの要領で描画ができる LINE－（X，Y），PSET として，出発点を省略した命令の使い方をしました。しかしこのままですと，いったん線を切って新たな位置から線を描かせたいときに，線が切れずに困ってしまうため，描き始めを区別する細工が必要になってきます。

このプログラムは描き始めを，READ　X で X 座標値を読んだときに，その数値が 1000 以上の場合に区別されるようにしてあります。つまり，実際の X 座標値に 1000 を加算したデー

（リスト 3-5）

```
 90 '********* DATA READ **********
100 READ X:IF X=  999 THEN 180
110 READ Y:IF X>=1000 THEN 150
120 '--------------------------------
130 LINE-(X,Y),PSET,7:GOTO100
140 '--------------------------------
150 X0=X-1000:Y0=Y:READ X,Y
160 LINE(X0,Y0)-(X,Y),PSET,7:GOTO 100
170 '--------------------------------
180 PAINT (150,60),2,7
190 PAINT (340,60),1,7:END
200 '--------------------------------
210 DATA 1130,50,270,50,270,70,170,70
220 DATA 170,80,250,80,250,100,170,100
230 DATA 170,130,130,130,130,50
240 DATA 1320,50,360,50,410,95,460,50
250 DATA 500,50,500,130,460,130,460,85
260 DATA 410,130,360,85,360,130,320,130
270 DATA 320,50,999
```



（写真 3-5　リスト 3-4, 3-5 の実行画面）

タが，描き始めを意味するわけです。

110 行で，X が 1000 以上だと判断されると 150 行に飛び，1000 を引いて X の値を本来の座標値に戻し，その数値を X0 に代入します。Y0 に Y の値を代入したのち，データ文から新しいデータを X，Y に読み込ませ，160 行で (X0, Y0)－(X, Y) として描画しています。

データ文最後の 999 は，READ 命令のストッパーの役目を果たしています。このストッパーがないと，データが無くなっても，なお，データを読み込もうとし，out of data エラーとなってプログラムは止まってしまいます。100 行で X の値が 999 であると判断されると 180 行に飛び，PAINT を実行し絵は完成されます。

● 配列に格納

メモリ効率が良く，カセットやディスクからの入出力も容易ですが，データの入力用プログラムが必要です。

100～140 行が入力用，150 行以降が描画用のルーチンです。リストが長くなるのを避けるために最小限の機能にとどめていますが，実用性をもたせるためには，もっとしっかりしたプログラムにしなければなりません。入力ルーチンは，もっと親切に設計する必要があるし，描画方法ももう少し工夫がいります。

（リスト 3-6）

```
 90 '********** DIMENSION **********
100 INPUT"ﾑｽﾌﾞ ﾃﾝ ﾉ ｶｽﾞ ";A
110 DIM X(A),Y(A)
120 FOR I=1 TO A:PRINT "No";I;
130 INPUT" X,Y ｻﾞﾋｮｳ ";X(I),Y(I)
140 NEXT I
150 '-----------------------------
155 LINE (X(1),Y(1))-(X(2),Y(2)),PSET
160 FOR I=3 TO A
180  LINE-(X(I),Y(I)),PSET
190 NEXT I
```

配列に格納したデータを描かせるプログラムの一例として，参考にとどめてください。

● メモリに直接格納

ひとつのデータを2バイトのメモリを使って，指定する番地以降に格納します。マシン語で，このデータを *BIOS に送り込み，描画させているのが普通です。

マシン語と聞くと，ずいぶん速く絵が描けそうな気がしますが，1本の線を引くスピードそのものは，BASICでもBIOSを利用したマシン語でもかわりません。線を引くルーチンはROMの中にマシン語で用意されており，BASICでLINE～と命令するとROM内のマシン語を呼びだして実行します。BIOS経由のマシン語でも結局同じことをしています。

マシン語は，データの転送や計算，ループに桁違いのスピードがあるものの，絵を描かせる全体の処理を通してBASICと比べてみると，その速度差はわずかなものです。

一概に，マシン語だからすべてが数十倍，数百倍速いとはいい切れないわけで，絵を描かせるプログラムでは，BASICでもマシン語に引け目を感ずる必要はありません。

(* BIOS → BASIC　Input Output System)

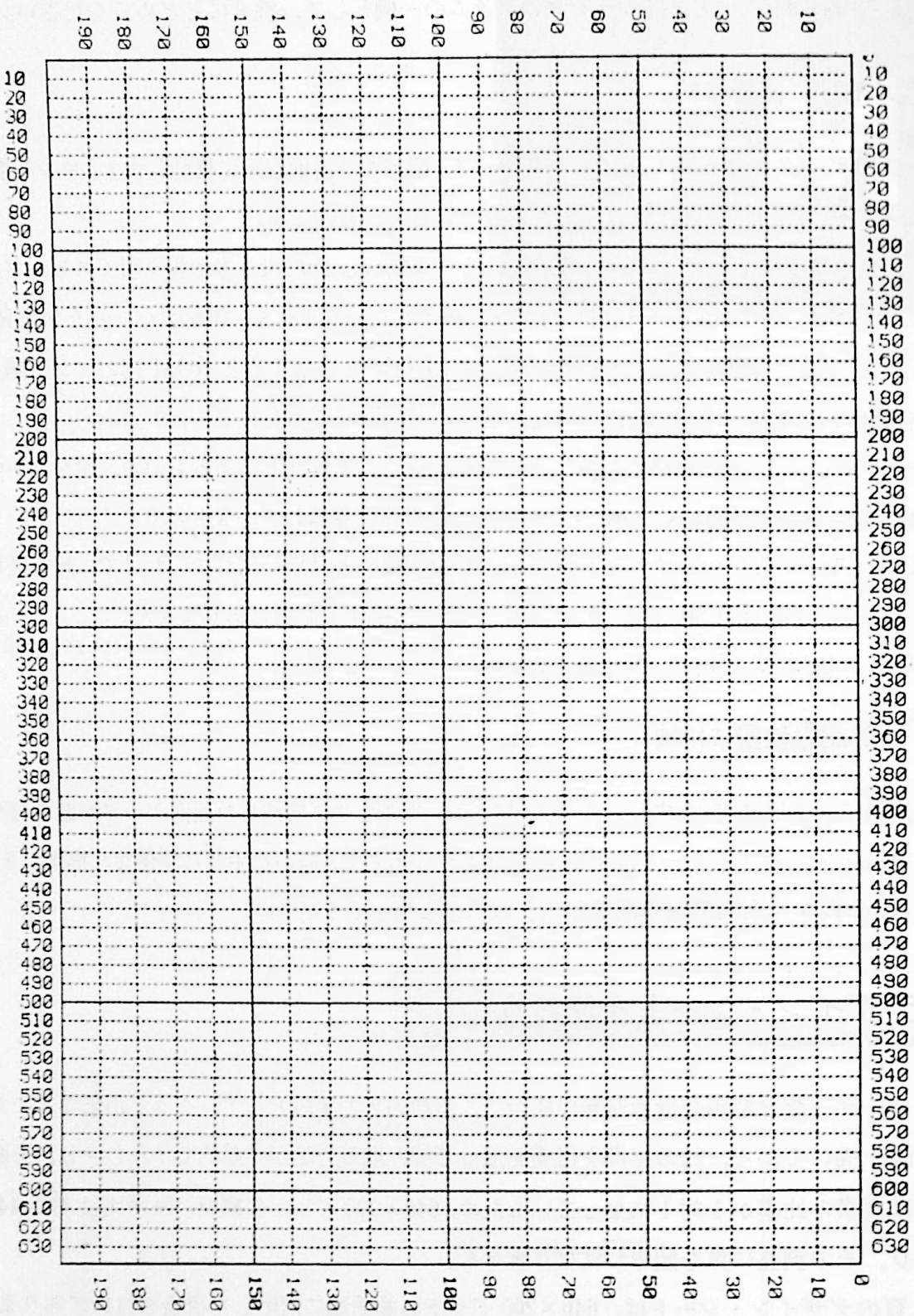
● 外部記憶装置に格納

少し意味あいが異なりますが，この方法はディスクなどに格納した画面データを読み描画します。プリンタに対するハードコピーと似たような出力で，描いた手順に関係なく描画します。詳しくは62頁の「画面情報の直接転送」の項を参照してください。

デザイン・シートを使おう

デジタイザーなど特別な機器が無い場合は，座標の入力手段として，グラフ用紙などの方眼紙に原画を描いてから，絵の座標を目盛を追って読み取る方法が一般的です。しかし，方眼紙の目盛は縦横の比率が1対1になっているため，640×200ドットの縦横の画面比(1対0.4495)と異なり，その調整に気を配らねばなりません。

46頁のデザイン・シートは，640×200ドットの画面比に対応した目盛を付けてあります。原画をこのシートに直接描き，その絵の座標を目盛から読み取れば，そのまま正確な座標デー


10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330 340 350 360 370 380 390 400 410 420 430 440 450 460 470 480 490 500 510 520 530 540 550 560 570 580 590 600 610 620 630
190 180 170 160 150 140 130 120 110 100 90 80 70 60 50 40 30 20 10 0

クとなります。何枚かコピーしておくと重宝すると思います。

スケッチボードプログラムの作成

これまで述べた方法は，あらかじめ絵のデータを作っておき，決められた手順に従って一気に描き上げてしまう方法をとりましたが，次にキーを使って画面上に直接描く方法について述べてみます。ディスプレイを画布に，キーボードを絵筆にみたてて，画面を見ながら絵を描きますので，なんとなく芸術的で楽しいプログラムになります。

直接，画面に絵を描くためには

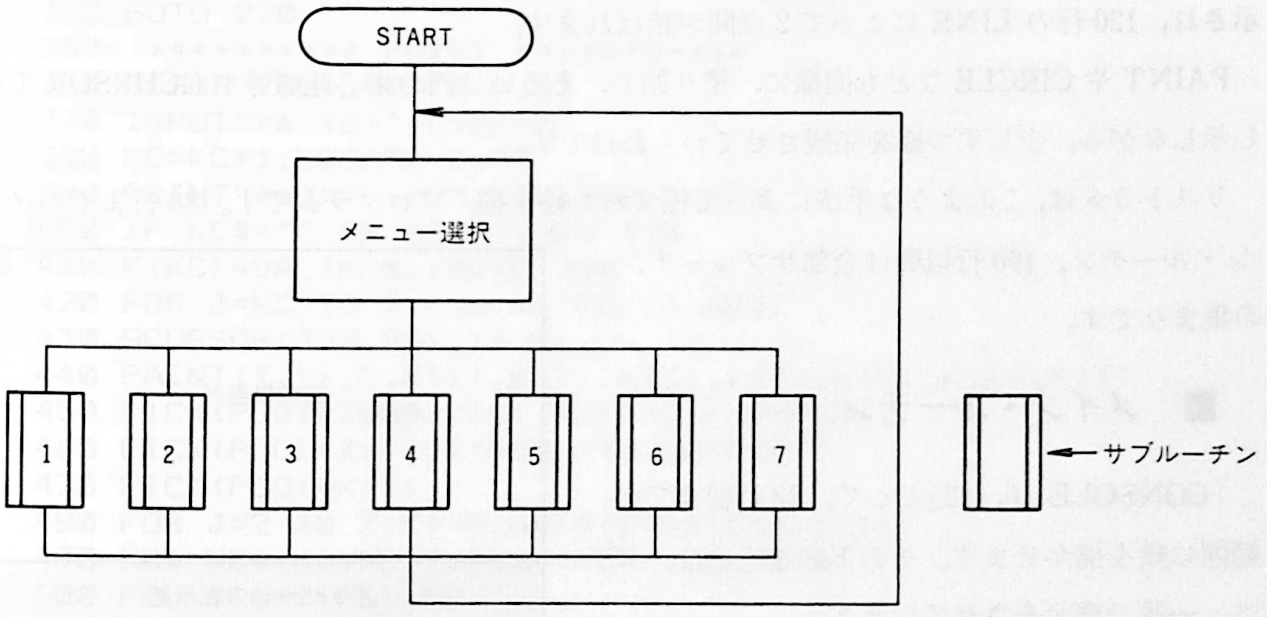
① 点を打つ，② 線を引く，③ 円を描く，④ 色を塗る

この４つの機能があれば，一応絵は描けますが，どうせならもう少し欲ばって

⑤ ブラシ表示，⑥ 画面の保存，⑦ 画面データの読み込み

の３つを追加して計７機能のプログラムを考えてみましょう。⑤のブラシ表示とはスプレイで絵を描く効果をねらったもので，指定する範囲内に指定する色のドットをランダムに描き込む機能を指します。

プログラムの構造は，これらの各機能をすべてサブルーチンにして，メニューの選択で呼び



(図3-1)

だす方法をとります。

このように，1機能＝1サブルーチンの形式にすると，プログラムはひとつずつ単独で考えることができ，また機能の追加や修正がとても楽になります。

■ グラフィックカーソル

絵筆にみたてるキー操作は，GCURSORが重要な役割を果たします。この命令については第2章ですでに述べてありますが，＋の形をしたGCURSORをカーソルキーで任意の位置に動かし，リターンキーを押すとその座標がドット単位で読めるというものです。

(リスト3-7)

```
100 GCURSOR(320,100),(X1,Y1),(X2,Y2)
110 PRINT X1;Y1,X2;Y2
120 LINE(X1,Y1)-(X2,Y2),PSET
```

このプログラムを実行すると，まず画面中央に＋のGCURSORが現れます。カーソルキーを使ってGCURSORを好きな所まで移動させ，リターンキーを押すとその位置のX，Y座標がX1とY1に代入されます。さらにカーソルキーでGCURSORを動かしリターンキーを押すと，その座標がX2とY2に代入されます。代入された各座標値は，110行のPRINT文で表示され，120行のLINEによって2点間が結ばれます。

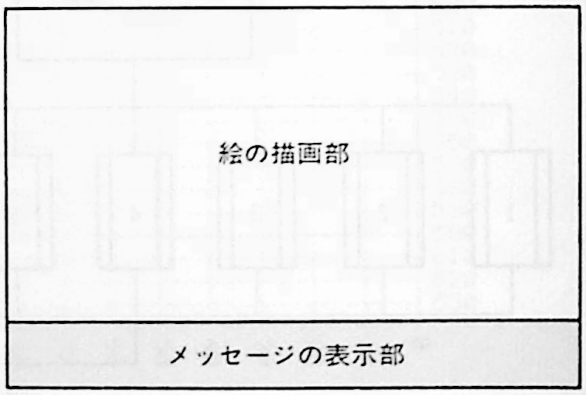
PAINTやCIRCLEなども同様に，塗り始め，あるいは円の中心座標等をGCURSORで指し示しながら，少しずつ絵を完成させていくわけです。

リスト3-8は，このような手法によって組まれた絵を描くプログラムです。180行までがメイン・ルーチン，190行以降は全部サブルーチンの集まりです。

■ メイン・ルーチン

「CONSOLE　0，22」として，22行目までの範囲に絵を描かせます。その下の3行には，メニュー等の表示をさせています。

このようにページモードを決めると，絵の描



(図3-2)

(リスト3-8　スケッチボードプログラム)

```
100 '********* エ ヲ カク プログラム ( スケッチ ボード ) **********
110 WIDTH 80,25:CONSOLE0,22:DIM PICX(2500),PICY(2500)
120 LINE(0,0)-(639,175),PSET,7,B
130 GOSUB 730:LOCATE 0,23:ON ERROR GOTO 180
140 PRINT"1=LINE   2=PAINT   3=BRUSH   4=CIRCLE";
150 PRINT"   5=SAVE   6=LOAD    0=END ":INPUT A
160 IF A=0 THEN WIDTH80:END
170 ON A GOSUB 190,350,510,630,1150,770:GOTO 130
180 CLOSE:BEEP:RESUME 130
190 '********** LINE **********
200 GOSUB 730:LOCATE 0,23:INPUT"セン ノ イロ ";C
210 PRINT"ﾂﾂﾞケル=CURSOR KEY+RETURN KEY";
220 PRINT"   オワル=RETUEN KEY+RETURN KEY";
230 GCURSOR(320,85),(X2,Y2),C
240 PSET(X2,Y2,C)
250 PICX(PCO)=1000:PICY(PCO)=C:PCO=PCO+1
260 PICX(PCO)=X2:PICY(PCO)=Y2:PCO=PCO+1
270 FOR T=0 TO 100:A$=INKEY$
280 IF A$=CHR$(13) THEN RETURN
290 NEXT
300 X1=X2:Y1=Y2
310 GCURSOR(X1,Y1),(X2,Y2),C
320 LINE(X1,Y1)-(X2,Y2),PSET,C
330 PICX(PCO)=X2:PICY(PCO)=Y2:PCO=PCO+1
340 GOTO 270
350 '********** PAINT **********
360 GOSUB 730:LOCATE 0,23
370 INPUT"ヌル イロ=";C:KC=0
380 KC=KC+1:LOCATE 0,23
390 PRINT"キョウカイ ショク";KC;:INPUT KC$
400 IF KC$="" OR KC=8 THEN 420
410 K(KC)=VAL(KC$):GOTO 380
420 FOR J=KC TO 7:K(J)=K(KC-1):NEXT
430 GCURSOR(320,85),(X,Y),C
440 PAINT(X,Y),C,K(1),K(2),K(3),K(4),K(5),K(6),K(7)
450 PICX(PCO)=2000:PICY(PCO)=C:PCO=PCO+1
460 PICX(PCO)=X:PICY(PCO)=Y:PCO=PCO+1
470 PICX(PCO)=K(1)
480 FOR J=2 TO 7:K$=RIGHT$(STR$(K(J)),1)
490 KC$=KC$+K$:NEXT:PICY(PCO)=VAL(KC$)
500 PCO=PCO+1:RETURN
510 '********* BRUSH **********
```

```
520 GOSUB 730:LOCATE 0,23
530 INPUT"ﾌﾞﾗｼ ﾉ ｲﾛ=";C
540 GCURSOR(320,85),(X1,Y1),(X2,Y2),C
550 FOR B=X1 TO X2 STEP SGN(X2-X1)
560 P=ABS(Y1-Y2):FOR I=0 TO P/5
570 IF Y1>Y2 THEN Y=Y2 ELSE Y=Y1
580 PSET(B,RND*P+Y,C):NEXT I,B
590 PICX(PC0)=3000:PICY(PC0)=C:PC0=PC0+1
600 PICX(PC0)=X1:PICY(PC0)=Y1:PC0=PC0+1
610 PICX(PC0)=X2:PICY(PC0)=Y2:PC0=PC0+1
620 RETURN
630 '************ CIRCLE ************
640 GOSUB 730:LOCATE 0,23
650 INPUT"ｴﾝ ﾉ ｲﾛ=";C:'LOCATE 0,23
660 PRINT"+ ﾖ ｳｺﾞｶｼ ﾁｭｳｼﾝ ﾄ ﾊﾝｹｲ ﾖ ｾｯﾄ ｼﾃｸﾀﾞｻｲ";
670 GCURSOR(320,85),(CX,CY),(HX,HY),C
680 CIRCLE(CX,CY),ABS(CX-HX),C
690 PICX(PC0)=4000:PICY(PC0)=C:PC0=PC0+1
700 PICX(PC0)=CX:PICY(PC0)=CY:PC0=PC0+1
710 PICX(PC0)=ABS(CX-HX):PC0=PC0+1
720 RETURN
730 '---------- ｶﾗｰ ﾁｬｰﾄ ------------
740 LOCATE 0,22:PRINT" 0  ";:FOR C=1 TO 7
750 COLOR C+8:PRINT C;"■";:NEXT C:COLOR7
760 PRINT"[STEP";PC0;"]":PRINTSPACE$(159);:RETURN
770 '****************** P LOAD ******************
780 INPUT"FILE NAME ";N$
790 PC0=0:OPEN"I",#1,N$:INPUT#1,PCL
800 FOR PL=0 TO PCL
810 INPUT#1,PICX(PL):INPUT#1,PICY(PL)
820 NEXT:PICX(PCL)=9999:CLOSE#1
830 CLS:LINE(0,0)-(639,175),PSET,7,B
840 IF PICX(PC0)=9999 THEN RETURN
850 IF PICX(PC0)=1000 THEN C=PICY(PC0):GOSUB 910
860 IF PICX(PC0)=2000 THEN C=PICY(PC0):GOSUB 960
870 IF PICX(PC0)=3000 THEN C=PICY(PC0):GOSUB 1020
880 IF PICX(PC0)=4000 THEN C=PICY(PC0):GOSUB 1100
890 PC0=PC0+1:GOTO 840
900 '---------------- LINE --------------------
910 PC0=PC0+1:IF PICX(PC0+1)>=1000 THEN 950
920 X1=PICX(PC0):Y1=PICY(PC0)
930 X2=PICX(PC0+1):Y2=PICY(PC0+1)
940 LINE(X1,Y1)-(X2,Y2),PSET,C:GOTO 910
```

```
 950 PSET(PICX(PCO),PICY(PCO),C):RETURN
 960 '------------------ PAINT --------------------
 970 X=PICX(PCO+1):Y=PICY(PCO+1)
 980 K(1)=PICX(PCO+2):FOR J=2 TO 7
 990 K(J)=VAL(MID$(STR$(PICY(PCO+2)),J,1)):NEXT J
1000 PAINT(X,Y),C,K(1),K(2),K(3),K(4),K(5),K(6),K(7)
1010 PCO=PCO+2:RETURN
1020 '------------------ BRUSH --------------------
1030 X1=PICX(PCO+1):Y1=PICY(PCO+1)
1040 X2=PICX(PCO+2):Y2=PICY(PCO+2)
1050 FOR B=X1 TO X2 STEP SGN(X2-X1)
1060 P=ABS(Y1-Y2):FOR I=0 TO P/5
1070 IF Y1>Y2 THEN Y=Y2 ELSE Y=Y1
1080 PSET(B,RND*P+Y,C):NEXT I,B
1090 PCO=PCO+2:RETURN
1100 '------------------ CIRCLE -------------------
1110 X1=PICX(PCO+1):Y1=PICY(PCO+1)
1120 X2=PICX(PCO+2)
1130 CIRCLE(X1,Y1),X2,C
1140 PCO=PCO+2:RETURN
1150 '******************* PSAVE **********************
1160 INPUT"FILE NAME ";N$
1170 OPEN"O",#1,N$:PRINT#1,PCO
1180 FOR PL=0 TO PCO
1190 PRINT#1,PICX(PL):PRINT#1,PICY(PL)
1200 NEXT:CLOSE#1:RETURN
1210 FOR I=0 TO PCO:PRINTPICX(I),PICY(I):NEXT
```

画部はスクロールの影響を受けなくなり安定した画面を得ることができます。

各機能のメニュー表示は,

1=LINE　線を引く。(点を打つはこの機能に含まれます)

2=PAINT　色を塗る。

3=BRUSH　指定する範囲に点の集まりを描く。

4=CIRCLE　円を描く。

5=SAVE　画面に描いた絵の座標データを記録する。

6=LOAD　絵の座標データを読み込み，描画する。

0=END　プログラムの実行を終了する。

以上の7項目で，項目名の頭にある0～6の数字を入力すると，ON　A　GOSUB 文によって対応するサブルーチンを呼びます。

■ (1) LINE

1を入力すると最初に線の色を求めてきます。上部のカラーチャートから0～7の色を選び，その数字を入力すると，画面中央に指定した色の＋（GCURSOR）が現れます。次に画面の下に次のような表示が出ます。

ツヅケル＝CURSOR　KEY＋RETURN　KEY

オワル　＝RETURN　KEY＋RETURN　KEY

任意の位置まで＋を動かし，リターンキーを押すとその位置が始点となり，指定色の点が表示されます。0.5秒ほどで，点は＋の表示に変わりますので，再び＋を動かし任意の位置でリターンキーを押してください。＋は消え始点からその位置まで線が引かれます。0.5秒ほどたつとまた＋がその位置に現れますので，同様の操作を繰り返し，次々と線をつなげ絵を描いていきます。

＋が現れるまで0.5秒のタイミングをとっていますが，これは続けるか終わるかの判断を待っているためで，＋が表示される前に再度リターンキーを押せば，LINE の実行は終了しメニューに戻ります。つまり，カーソルキーで＋を移動させた後，リターンキーをポンポンと2回たたくと線の描画はそこで終了し，1回しか押さなければ続行するという仕組みです。

■ (2) PAINT

塗る色と境界色を求めてきますので，それぞれ入力してください。境界色は7つまで指定できます。

例えば2つの同心円ではさまれたドーナツ状の範囲を塗る場合を説明します。内側の円は赤で，外側の円は黄色で描かれているものとし，これらの円の色を境界色とし，中を紫で塗るとします。

ヌルイロ＝？　で3↵とします。

キョウカイショク1？＝　と出ます。1本目の境界色を聞いていますので赤ですから2↵とします。

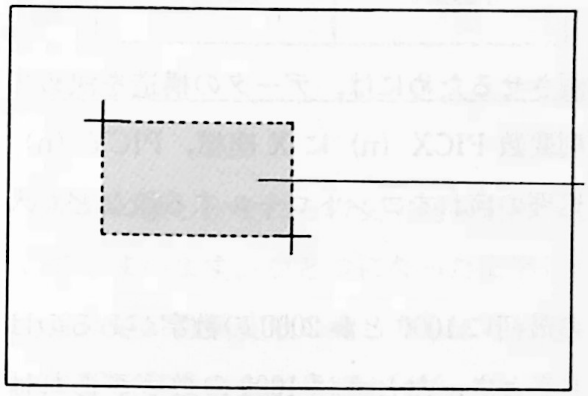
キョウカイショク2?＝　と2本目の境界色を聞いてきますので，黄色ですから6↵です。

キウカイショク3?＝　と聞いてきますが3本目は必要ないので↵のみ押します。これで境界色の入力は終わりです。その後＋が画面中央に現れますので，塗るべき領域まで＋を移動させリターンキーを押すと PAINT が実行されます。

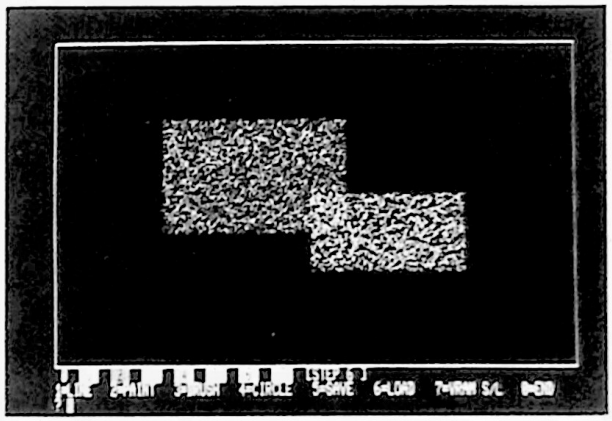
PAINT は細心の注意を払って実行してください。領域を囲む境界線が完全に閉じられていないと，せっかく描いた絵が塗り消されてしまいます。また境界色の指定も同様の意味で慎重に行なう必要があります。

■ (3) BRUSH

ブラシの色を入力すると，画面中央に＋が表示されます。ブラシの範囲を＋を移動させ2点を指定しますが，2点間の対角線を持つ四角の中がその範囲となります。つまり，LINE 命令の B(BOX) 指定と同じです。2点間を指定しリターンキーを押すと，その範囲が指定色のドットで埋められます。ドットの密度は560行の P/5 の値を変更することにより変えられます。



(図 3-3)



(写真 3-6　ブラシの作画例)

■ (4) CIRCLE

円の色を入力すると＋が現れます。円の中心となる位置まで＋を動かしてリターンキーを押し，次に半径（円の外周）位置に＋を移動させリターンキーをたたくと指定した円が描画できます。半径の指定はX（横）方向で行ないます。

■ (5) SAVE

これまで描いた絵のデータをテープやディスクに記録します。ファイル名を入力しリターンキーを押すと記録を開始します。なお，ファイル名は8文字以内にしてください。

■ (6) LOAD

テープやディスクから絵のデータを読み込み描画します。セーブした時のファイル名を入力し，リターンキーを押します。

● データの構造

絵のデータを記録したり，あるいは読み込んで描画させるためには，データの構造を決めておかなければなりません。このプログラムでは，配列変数PICX（n）にX座標，PICY（n）にY座標を格納させることを原則にしていますが，処理の流れをコントロールする値なども入れてあります。

図3-4は各機能ごとのデータの構造です。データの最初に1000とか2000の数字があるのは機能の区分のためです。データを読み込んで描画するとき，たとえば1000の数字があればLINEのルーチンへ，2000があればPAINTのルーチンへと，この数字によって処理のルーチンが自動的に選ばれることになります。850～880行のIF文が，その処理をしています。（図3-5参照）

PAINT機能の境界色は少し変わったデータの格納方法をとっています。境界色の数は7つまで指定できるのですが，常に7つ分の配列を確保しておくことはメモリーが無駄になりますので，データの圧縮を試みました。圧縮は文字列を使用します。境界色の数値を，ひとつずつ

LINE

	PIC X(n)	PIC Y(n)
1	1000	カラー番号
2	X座標1(始点)	Y座標1(始点)
3	X座標2	Y座標2
⋮		

PAINT

	PIC X(n)	PIC Y(n)
1	2000	カラー番号
2	境界色1	境界色 2, 3, 4, 5, 6, 7
3	X座標	Y座標

BRUSH

	PIC X(n)	PIC Y(n)
1	3000	カラー番号
2	X座標1	Y座標1
3	X座標2	Y座標2

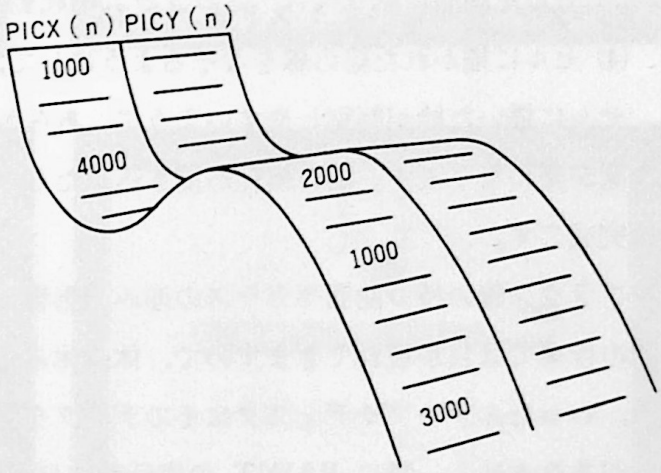
CIRCLE

	PIC X(n)	PIC Y(n)
1	4000	カラー番号
2	中心のX座標	中心のY座標
3	半径	——

(図3-4　データの構造)

文字列に直し，それらの文字列を全部つないでしまいます。ひとつになった文字列を，今度は逆に数値変数に直してしまうとデータの圧縮が出来あがりです(480, 490行)。このデータを読み込んで描画させるときは，逆のことを行なってひとつずつ境界色の数値を取り出します(980, 990行)。

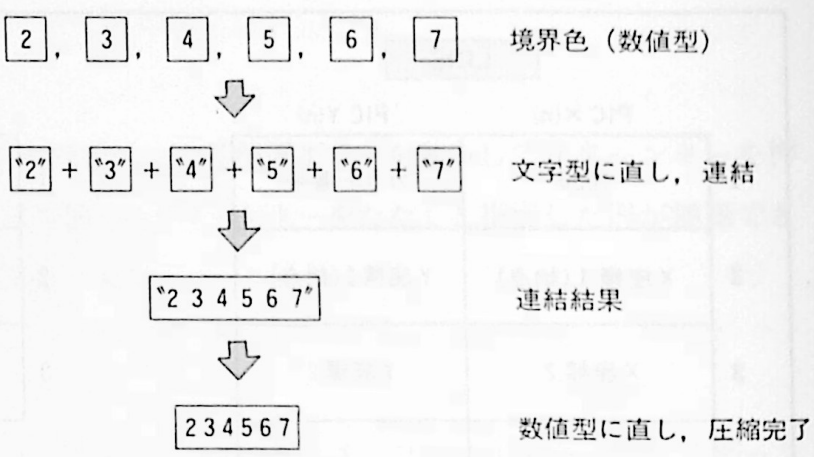
なお，データを格納する配列は PICX, PICY とも2500までとってあります。2500を超えるデータの格納はできません。また，厳密な入力チェックはしていませんので，



(図3-5　データの流れ)

とんでもない値を入れるとエラーが生じることがあります。エラーが起きたときは，一応，ON　ERROR GOTO 文によって強制的にメニューに戻しています。

プログラムを打ち込む際，完動が確認されるまでは，ON ERROR GOTO と，CONSOLE 文は取っておいたほうが無難です。



（図3-6 データの圧縮）

● 上手に絵を描くために

GCURSOR の＋を目で追いながら絵を描くプログラムはこれで完成しました。一応は何でも描くことは可能ですが，画面上に直接完成度の高い絵を描き上げることは，なかなか難しいものです。簡単で能率的な一方法を紹介しておきます。

(1) 使用ディスプレイの画面と同じ大きさの下絵を描く。

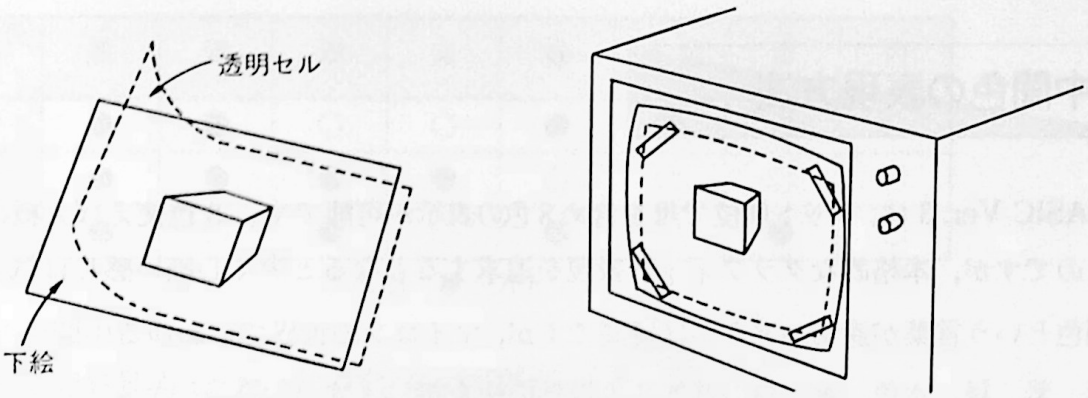
(2) 透明なセルに下絵を描き写す。

(3) セルを画面に貼り付け，プログラムを起動する。

(4) セルに描かれた絵の線をなぞるようにして，GCURSOR を動かし画面に絵を描く。

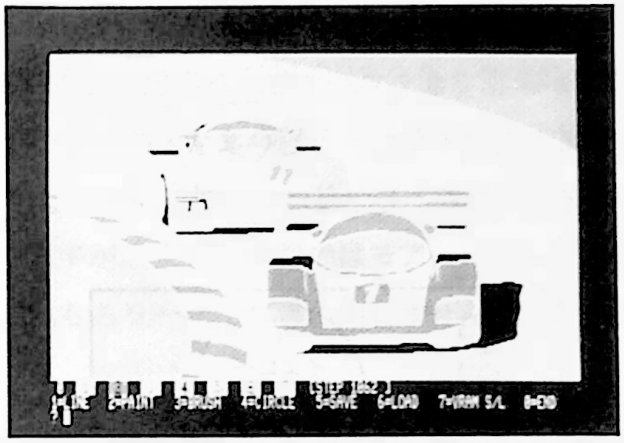
セルに描いた絵が判別しやすいように，あらかじめ画面全体を青などで PAINT しておくと作業が楽になります。絵の輪郭が描き込めたらセルをはがして，各領域の PAINT を実行すれば完成です。

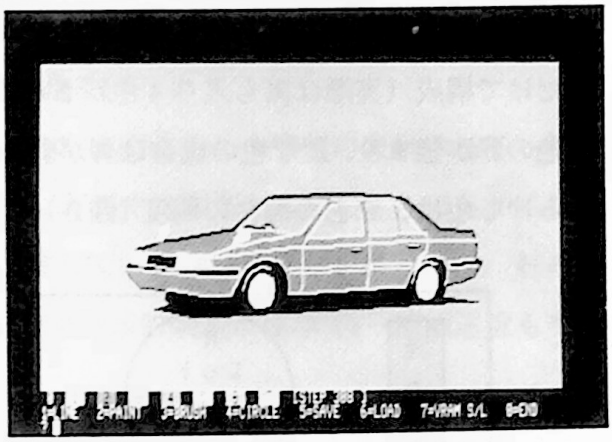
ブラウン管の持つ曲面やガラスの厚みの影響で，セルの絵と GCURSOR の両方を見る長時間の作業では目が疲れてきますので，休み休み行なってください。また，ある程度絵が描けたら，いったんテープやディスクにそのデータを記録してから，作業を続けるようにすると間違いがありません。特に PAINT の実行前には，SAVE しておいてからのほうが無難です。境界色をひとつ間違えたために，せっかく描いた絵がみるみるうちに塗り消されてしまっては後の祭りです。

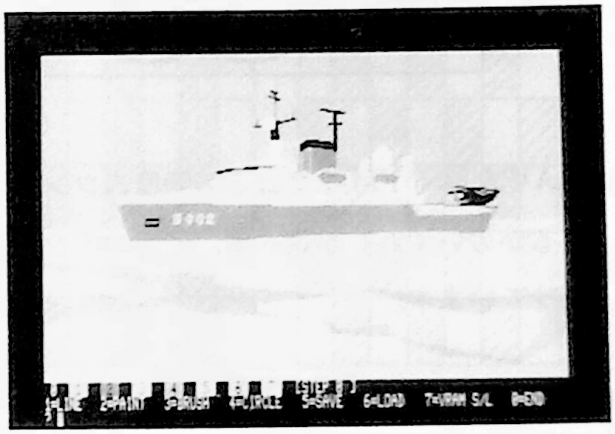


(図 3-7)

● スケッチボードプログラムで作成した絵







(写真 3-9)

(写真 3-10)

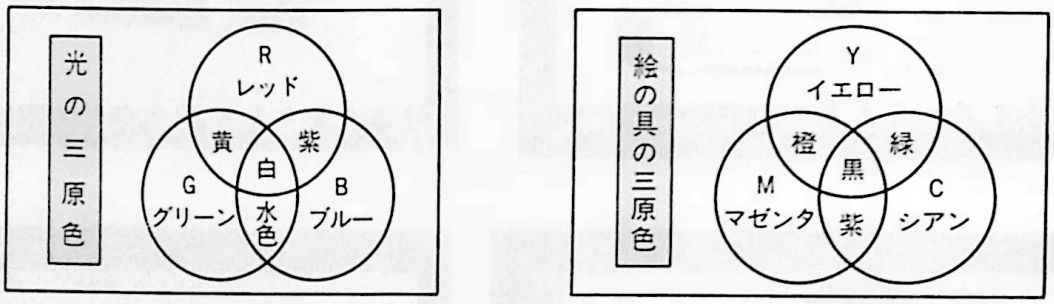
中間色の表現方法

F・BASIC Ver. 3 は，ドット単位で黒を含め8色の表示が可能です。8色使えば大概のことはできるのですが，本格的なグラフィック表現を追求するとなると少々心細い感じは拭えません。

中間色という言葉が適切かどうかは疑問ですが，マイコンの世界でいわゆる中間色とは，黒，青，赤，紫，緑，水色，黄，白の基本8色以外の色を指します。

パソコンでの中間色合成の具体例を述べる前に，三原色について考えてみましょう。三原色とは，あらゆる色の合成を行なううえで，基本となる3つの色を指し，〝光の三原色〟と〝絵の具の三原色〟の2つの系統があります。

光の三原色を使用したものに，カラーテレビのブラウン管。絵の具の三原色を使ったものに印刷のカラー写真などがあげられます。ブラウン管や印刷物をルーペで拡大してみると，三原色だけで構成（実際は黒も入り4色）されているのが確認できます。また，ある色のときは三原色の赤が強まり，別な色の場合は青が弱まっていることにも気付くと思います。このように，あらゆる色は三原色の個々の明度（強さ）を微妙に変化させることで表現することができます。



（図3-8　三原色）

AV を除く FM-7 ではこの明度のコントロールができません。三原色の各色は，輝くか消えるかのいずれしか選べず，三色のオン・オフの組み合わせ（2の3乗）で8色表示にとどまっています。

表示色 / 三原色	黒	青	赤	紫	緑	水色	黄	白
R	●	●	○	○	●	●	○	○
G	●	●	●	●	○	○	○	○
B	●	○	●	○	●	○	●	○

○…ON　●…OFF

2色の組み合わせで作る中間色

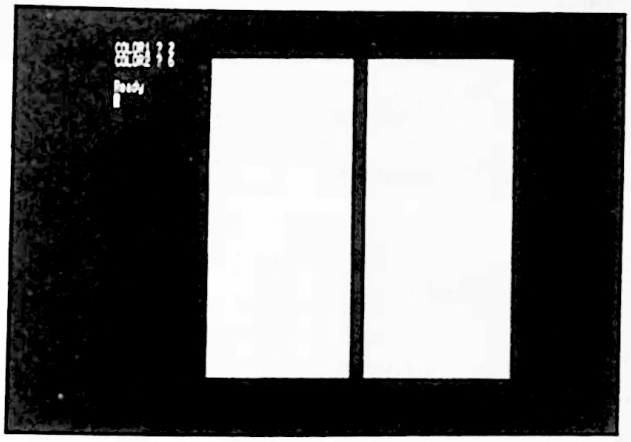
リスト 3-9 は，2 色の混合で中間色を合成するプログラムです。

COLOR1？

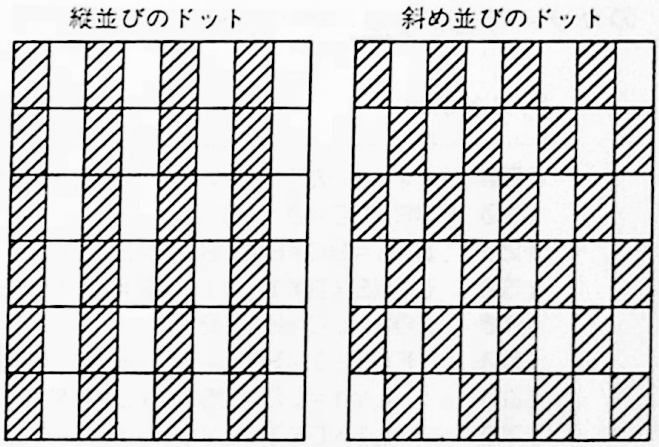
COLOR2？

と順に聞いてきますので，0～7 から任意のカラー番号を入力してください。COLOR1 の色で下地を塗ったあと，COLOR2 の色を割り込ませて中間色を作ります。

写真 3-11 は，このプログラムの出力画面です。左と右に同じ色の中間色が表示されていますが，左はドットを縦並び，右は斜めに並べています。写真では両者とも同じように見えると思いますが，実際の画面ではモアレの現れ方に違いが生じています。どちらが良いかは，好みの問題もあり一概には言い切れません。縦のほうはプログラムの構造が簡単で，描画速度もやや



(写真 3-11　縦と斜め並びの中間色表示)



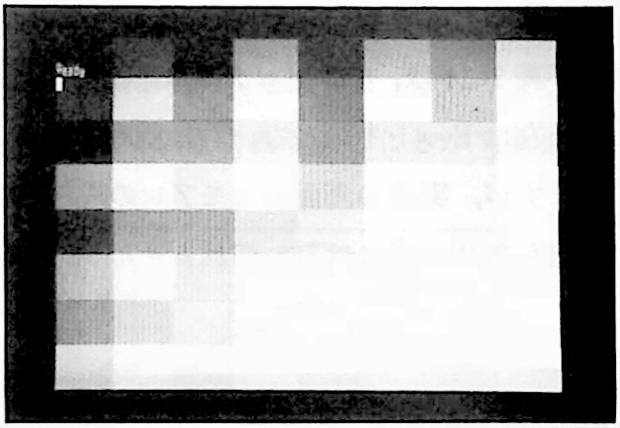
(図3-9)

速くなります。組み合わせる色の種類やディスプレイの解像度にもよりますが，3色以上の場合は斜めのほうが良いようです。

(リスト3-9　2色の混合)

```
100 '********** 2 ショク ノ コンゴウ **********
110 CLS
120 INPUT"COLOR1 ";C1
130 INPUT"COLOR2 ";C2
140 '-----------------------------------
150 LINE(130,10)-(310,190),PSET,C1,BF
160 LINE(330,10)-(510,190),PSET,C1,BF
170 '-----------------------------------
180 FOR I=0 TO 180 STEP 2
190  LINE(130+I,10)-(130+I,190),PSET,C2
200  LINE(330+I,10)-(330,10+I),PSET,C2
210  LINE(510,10+I)-(330+I,190),PSET,C2
220 NEXT
```

リスト3-10は，2色の組み合わせによる中間色のカラーチャート表示プログラムです。FOR～NEXTのループが3重なので，やや理解しづらいかも知れません。図3-10は各色の組み合わせ表です。写真3-12はこのチャートを示したのですが，実際の色は第1章のカラー写真を参照してください。



(写真3-12　カラーチャート)

(リスト3-10)

```
100 '**** カラー チャート / 2 ショク ノ コンゴウ *****
110 FOR BC=0 TO 7
120  BX1=BC*80:BX2=BX1+80
130  LINE(BX1,0)-(BX2,199),PSET,BC,BF
140  FOR LC=0 TO 7
150   FOR I=BX1 TO BX2 STEP 2
160    LY1=LC*25:LY2=LY1+25
170    LINE(I,LY1)-(I,LY2),PSET,LC
180   NEXT I
190  NEXT LC
200 NEXT BC
```

0—0	1—0	2—0	3—0	4—0	5—0	6—0	7—0
0—1	1—1	2—1	3—1	4—1	5—1	6—1	7—1
0—2	1—2	2—2	3—2	4—2	5—2	6—2	7—2
0—3	1—3	2—3	3—3	4—3	5—3	6—3	7—3
0—4	1—4	2—4	3—4	4—4	5—4	6—4	7—4
0—5	1—5	2—5	3—5	4—5	5—5	6—5	7—5
0—6	1—6	2—6	3—6	4—6	5—6	6—6	7—6
0—7	1—7	2—7	3—7	4—7	5—7	6—7	7—7

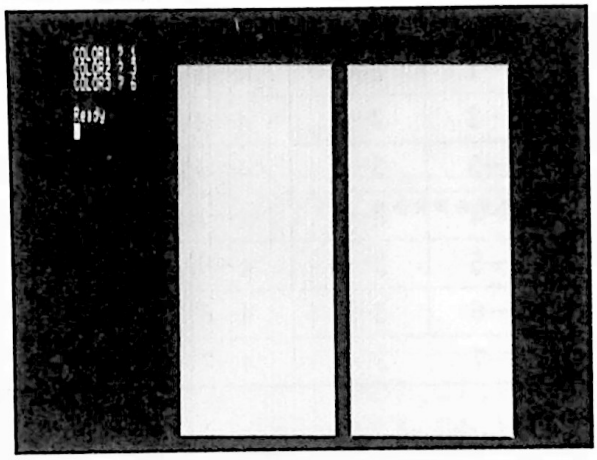
(図 3-10　カラーチャートの構成)

3色の組み合わせで作る中間色

3色使うと表現できる中間色の種類はグンと豊富になります。リスト 3-11 は，3色で表現する中間色プログラムです。実行方法や画面表示は，リスト 3-6 の 2 色の場合とまったく同じです。さまざまな組み合わせを入力し，微妙な色あいを楽しんでください。

(リスト 3-11)

```
100 '********** 3 ショク ノ コンゴウ **********
110 CLS
120 INPUT"COLOR1 ";C1
130 INPUT"COLOR2 ";C2
135 INPUT"COLOR3 ";C3
140 '-------------------------------
150 LINE(130,10)-(310,190),PSET,C1,BF
160 LINE(330,10)-(510,190),PSET,C1,BF
170 '-------------------------------
180 FOR I=0 TO 180 STEP 3
190  LINE(130+I,10)-(130+I,190),PSET,C2
195  LINE(130+I+1,10)-(130+I+1,190),PSET,C3
200  LINE(330+I,10)-(330,10+I),PSET,C2
205  LINE(330+I+1,10)-(330,10+I+1),PSET,C3
210  LINE(510,10+I)-(330+I,190),PSET,C2
215  LINE(510,10+I+1)-(330+I+1,190),PSET,C3
220 NEXT
```



(写真 3-13)

画面情報の直接転送法とスケッチボード 2

画面に表示されている図形などの情報をディスクに直接格納する一方法を紹介します。スケッチボードプログラムでは，絵の座標データ等を転送しましたが，今度の方法は，V-RAM をそっくりディスクにコピーしてしまいます。つまり，描いた手順によらず，画面上部から1ラインずつ転送し，画面を記録します。描画するときも同様に，ディスクから読み込みながら1ラインずつ表示していきます。

当初，テープを使っての転送も試みましたが，速度が遅過ぎて実用になりませんでした。画面全体で 48 K バイトもあるわけですから無理もありません。テープ使用の場合は，座標のデータを転送する方法にしたほうがずっと効率的です。

さて，V-RAM をコピーすると書きましたが，これはあまり正確な表現ではありません。

V-RAM はサブ CPU が直接管理しています。ベーシックからでは，その値をのぞきようがありませんので，GET ⓐ，PUT ⓐ命令を使用して，画面情報を転送することにしました。つまり，画面を GET ⓐした配列をディスクに記録し，描画するときは反対にその配列を読み込んで PUT ⓐすれば良いことになります。ただし，画面全体を一度に GET ⓐして配列に入れるわけにはいきません。ベーシックで使える 30 K バイトあまりのメモリーを全部配列の確保に回しても足りなくなります。ここでは，横1ライン 640 ドット分ずつ配列に入れる方法をとりました。こうすると配列は整数型で 120 個とれば良いことになりますので，無駄なメモリーを

使わなくてすみます。

■ 転送の実験

最初に，リスト3-12のプログラムを考えてみました。130行のGET ⓐで，1ライン分のデータを配列E%に格納し，140～160行のFOR文でE%(0)～E%(120)までのそれぞれの値をディスクに書き込んでいます。220行のFOR文は，画面上部から1ラインずつ199ラインまでのループです。

リスト3-13は，リスト3-12で書き込んだデータを読み取り，画面に出力するためのプログラムです。

(リスト3-12)

```
100 DIM E%(120)
110 OPEN "O",#1,"VRAM"
120 FOR Y=0 TO 199
130  GET@A(0,Y)-(639,Y),E%,G
140  FOR I=0 TO 120
150   PRINT#1,E%(I)
160  NEXT I
170 NEXT Y
180 CLOSE#1:END
```

(リスト3-13)

```
200 DIM E%(120)
210 OPEN "I",#1,"VRAM"
220 FOR Y=0 TO 199
230  FOR I=0 TO 120
240   INPUT#1,E%(I)
250   PUT@A(0,Y)-(639,Y),E%,PSET
260  NEXT I
270 NEXT Y
280 CLOSE#1:END
```

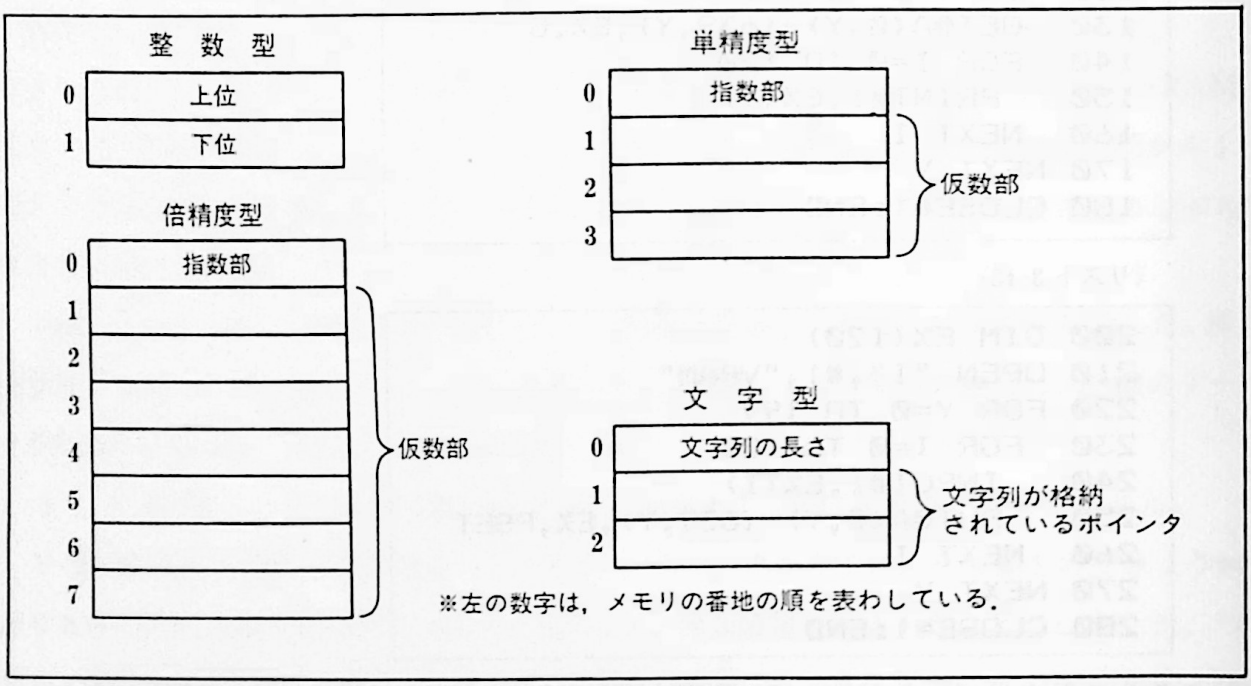
このリストを見ていただければ，画面情報の基本的な転送アルゴリズムは理解してもらえると思いますが，正直な話，このプログラムは使いものになりません。ディスクを使っても，昼寝ができそうなくらい転送速度が遅いのです。一番のネックは，120個の配列転送のループです。1ラインずつ実行するためのループが200回ありますので，2つのループでは24,000回も通すことになりますので，時間がかかるのもうなづけます。

■ マシン語の助けをかりる

120個の配列をバラバラにせず，一度に転送できれば，単純計算で120倍のスピードアップができます。そのためのデータの圧縮には文字列を使うことが考えられます。しかし，ベーシックで120個のデータを全部つなげてひとつの文字列に直すためには，それだけでかなりの時間がかかってしまいます。ここはマシン語の助けをかりなければならないようです。

■ 変数の格納法を知る

マシン語で，配列に入っている複数の数値を，1つの文字列に置き換えるためには，F-BASIC自体の中身について知る必要があります。特に，各型の変数がどのようにメモリーに蓄えられているかを理解することが重要です。



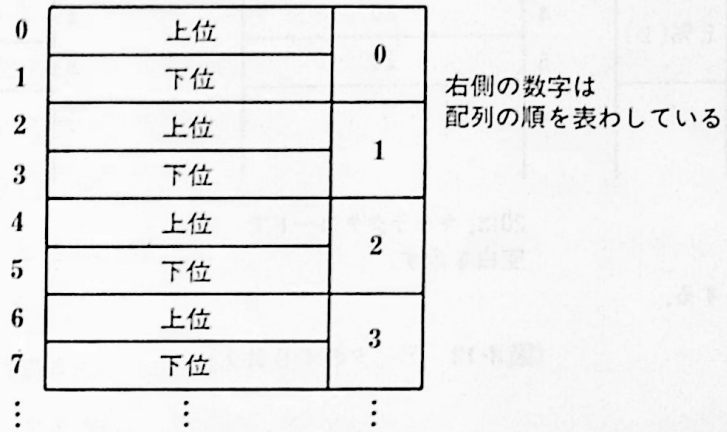
(図3-11　データの格納形式)

図3-11は，データの格納形式を示したものです。VARPTR（バリアブルポインタ）で指し示めされるアドレスは，この図の0の位置になります。VARPTRは変数を格納している番地を与える命令です。

整数型の変数は，ひとつの値を格納するために2バイトのメモリーを使用し，上位と下位に

ふり分けています。これは，1バイトでは256までの数値しか扱えないためで，2バイト使うことにより，はじめて＋32767～－32768までの値を入れることができるからです。

整数型の配列の場合は，図3-12のようになっています。データは配列順に従って2バイトずつ整然と並んでおり，DIM 文で宣言した配列の数だけメモリーが確保されています。



（図3-12　整数型配列の格納形式）

■ データをすり替える

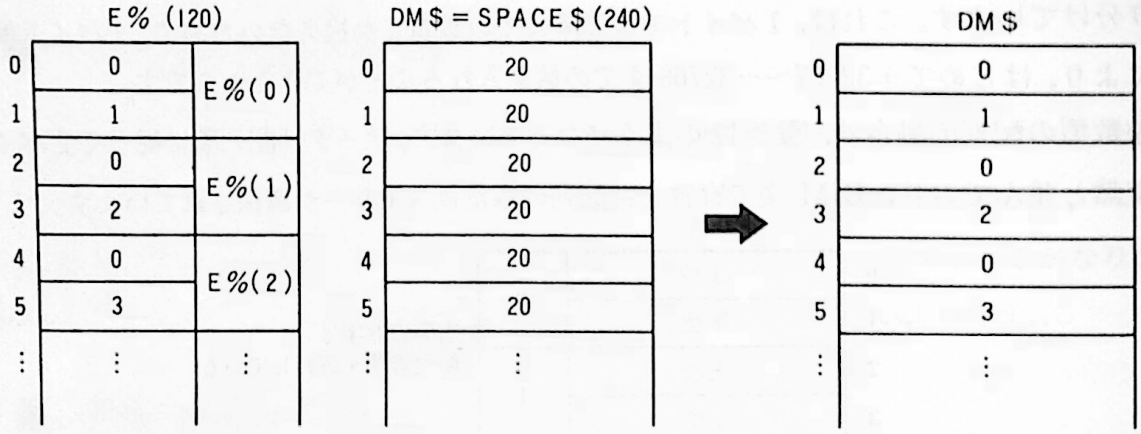
さて，これで整数型の配列がどのようにメモリーに格納されているかがわかりました。データ圧縮のために，整数型配列の各データを全部まとめてひとつの文字列にしてしまうためには，メモリー上に展開されたデータの並びを利用します。ここで，

```
DIM　E％（120）
DM$＝SPACE$（240）
```

と設定すると配列変数E％には120個分の配列（正確には0を含めて121個）が用意され，240バイトのメモリーがE％のために確保されます。また，文字変数のDM$には240個の空白が代入され，同じく240バイトのメモリーが確保できます。

データのすり替えは，E％の配列が展開されているメモリーを，DM$のために確保されているメモリー空間に転送してしまうことで可能になります。（図3-13）

このメモリーの転送を，ベーシックのPEEK，POKE　を使って行なうのでは時間がかかり過ぎてしまうので，この処理だけはマシン語にまかせたほうが無難です。処理は240回のループとメモリーの転送だけですので，マシン語としては最も簡単な部類に入ります。



(図3-13　データのすり替え)

(リスト3-14)

```
0001  0000                ;++++++++++++++++++++
0002  0000                ;   240 ﾊﾞｲﾄ ﾉ ﾃﾝｿｳ
0003  0000                ;++++++++++++++++++++
0004  6B04                        ORG   $6B04
0005  6B04                ;
0006  6B04 C6F0                   LDB   #$F0
0007  6B06 BE6B00                 LDX   $6B00
0008  6B09 10BE6B02               LDY   $6B02
0009  6B0D                ;
0010  6B0D A680           LOOP:   LDA   ,X+
0011  6B0F A7A0                   STA   ,Y+
0012  6B11 C001                   SUBB  #1
0013  6B13 26F8                   BNE   LOOP
0014  6B15 39                     RTS

add   +0 +1 +2 +3 +4 +5 +6 +7 sum
6B00  00 00 00 00 C6 F0 BE 6B :DF
6B08  00 10 BE 6B 02 A6 80 A7 :08
6B10  A0 C0 01 26 F8 39 00 00 :B8
6B18  00 00 00 00 00 00 00 00 :00
6B20  00 00 00 00 00 00 00 00 :00
6B28  00 00 00 00 00 00 00 00 :00
6B30  00 00 00 00 00 00 00 00 :00
6B38  00 00 00 00 00 00 00 00 :00
```

本書ではマシン語の説明は省きますが，念のためアセンブルしたソースリストとダンプリストをリスト 3-14 に載せておきます。

■ 画面転送プログラムの実際

これまで述べたことをベースにプログラムを組み，これをスケッチボードプログラム（リスト 3-8）に合体させたのがリスト 3-15 です。9000 行以降が V-RAM コピーのサブルーチンで，9060～9140 行がディスクへの書き込み用，9160～9250 行が読み込みと描画ルーチンです。このサブルーチンは，80 行，90 行とセットになっています。自分で作ったプログラムに，このサブルーチンを連結するときは，80 行，90 行の内容をプログラムの先頭に入れ「GOSUB 9000」として活用してください。

なお，マシン語は 9270 行，9280 行に DATA として格納しています。RUN すると 90 行の READ 文でこのデータを読み，POKE 文で直接メモリーに書き込みます。また，マシン語プログラムは &H6B00 番地以降に展開していますので，CLEAR 文でユーザーエリアを制限しておかねばなりません（80 行）。

リスト 3-15 は，リスト 3-8 のスケッチボードプログラムの内容と同じですが，メニューに (7＝VRAM S/L) を追加しました。メニューで 7 を選ぶと SAVE か LOAD か聞いてきます。入力後，ファイル名を入れると画面の転送を開始します。なお，メモリーの関係でスケッチボード 2 で作画できるステップ数は 2000 までとなります。

フロッピー頒布サービス

正確に入力する自信のない方や，入力する時間のない多忙な方のために，本書収載の全プログラムをフロッピーディスクに収録して頒布しております。詳細は目次前にありますので，ご覧のうえお申し込みください。

(リスト3-15　スケッチボードプログラム2)

```
page 1    [ File Name : SKETCH ]
```

```
 60 '◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆ ｽｹｯﾁ ﾎﾞｰﾄﾞ 2 ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
 70 '
 80 CLEAR600,&H6AFF:DIM E%(120)
 90 FOR I=0 TO 17:READ PD:POKE &H6B04+I,PD:NEXT
100 WIDTH 80,25:CONSOLE0,22:DIM PICX(2000),PICY(2000)
110 LINE(0,0)-(639,175),PSET,7,B
120 GOSUB 710:LOCATE 0,23:ON ERROR GOTO 165
130 PRINT"1=LINE   2=PAINT   3=BRUSH   4=CIRCLE";
140 PRINT"   5=SAVE   6=LOAD   7=VRAM S/L   0=END ":INPUT A
150 IF A=0 THEN WIDTH80:END
160 ON A GOSUB 170,330,490,610,1380,1000,9000:GOTO 120
165 BEEP:CLOSE:RESUME 120
170 '********* LINE **********
180 GOSUB 710:LOCATE 0,23:INPUT"ｾﾝ ﾉ ｲﾛ ";C
190 PRINT"ﾂﾂﾞｹﾙ=CURSOR KEY+RETURN KEY";
200 PRINT"   ｵﾜﾙ=RETUEN KEY+RETURN KEY";
210 GCURSOR(320,85),(X2,Y2),C
220 PSET(X2,Y2,C)
230 PICX(PC0)=1000:PICY(PC0)=C:PC0=PC0+1
240 PICX(PC0)=X2:PICY(PC0)=Y2:PC0=PC0+1
250 FOR T=0 TO 100:A$=INKEY$
260 IF A$=CHR$(13) THEN RETURN
270 NEXT
280 X1=X2:Y1=Y2
290 GCURSOR(X1,Y1),(X2,Y2),C
300 LINE(X1,Y1)-(X2,Y2),PSET,C
310 PICX(PC0)=X2:PICY(PC0)=Y2:PC0=PC0+1
320 GOTO 250
330 '********** PAINT **********
340 GOSUB 710:LOCATE 0,23
350 INPUT"ﾇﾙ ｲﾛ=";C:KC=0
360 KC=KC+1:LOCATE 0,23
370 PRINT"ｷｮｳｶｲ ｼｮｸ";KC;:INPUT KC$
380 IF KC$="" OR KC=8 THEN 400
390 K(KC)=VAL(KC$):GOTO 360
400 FOR J=KC TO 7:K(J)=K(KC-1):NEXT
410 GCURSOR(320,85),(X,Y),C
420 PAINT(X,Y),C,K(1),K(2),K(3),K(4),K(5),K(6),K(7)
430 PICX(PC0)=2000:PICY(PC0)=C:PC0=PC0+1
```

page 2　[File Name : SKETCH]

```
 440 PICX(PCO)=X:PICY(PCO)=Y:PCO=PCO+1
 450 PICX(PCO)=K(1)
 460 FOR J=2 TO 7:K$=RIGHT$(STR$(K(J)),1)
 470 KC$=KC$+K$:NEXT:PICY(PCO)=VAL(KC$)
 480 PCO=PCO+1:RETURN
 490 '******** BRUSH **********
 500 GOSUB 710:LOCATE 0,23
 510 INPUT"ﾌﾞﾗｼ ﾉ ｲﾛ=";C
 520 GCURSOR(320,85),(X1,Y1),(X2,Y2),C
 530 FOR B=X1 TO X2 STEP SGN(X2-X1)
 540 P=ABS(Y1-Y2):FOR I=0 TO P/5
 550 IF Y1>Y2 THEN Y=Y2 ELSE Y=Y1
 560 PSET(B,RND*P+Y,C):NEXT I,B
 570 PICX(PCO)=3000:PICY(PCO)=C:PCO=PCO+1
 580 PICX(PCO)=X1:PICY(PCO)=Y1:PCO=PCO+1
 590 PICX(PCO)=X2:PICY(PCO)=Y2:PCO=PCO+1
 600 RETURN
 610 '********** CIRCLE **********
 620 GOSUB 710:LOCATE 0,23
 630 INPUT"ｴﾝ ﾉ ｲﾛ=";C:'LOCATE 0,23
 640 PRINT"+ ヲ ｳｺﾞｶｼ ﾁｭｳｼﾝ ﾄ ﾊﾝｹｲ ヲ ｾｯﾄ ｼﾃｸﾀﾞｻｲ";
 650 GCURSOR(320,85),(CX,CY),(HX,HY),C
 660 CIRCLE(CX,CY),ABS(CX-HX),C
 670 PICX(PCO)=4000:PICY(PCO)=C:PCO=PCO+1
 680 PICX(PCO)=CX:PICY(PCO)=CY:PCO=PCO+1
 690 PICX(PCO)=ABS(CX-HX):PCO=PCO+1
 700 RETURN
 710 '---------- ｶﾗｰ ﾁｬｰﾄ ----------
 720 LOCATE 0,22:PRINT" 0  ";:FOR C=1 TO 7
 730 COLOR C+8:PRINT C;"■";:NEXT C:COLOR7
 740 PRINT"[STEP";PCO;"]":PRINTSPACE$(159);:RETURN
1000 '****************** P LOAD ******************
1010 INPUT"FILE NAME ";N$
1020 PCO=0:OPEN"I",#1,N$:INPUT#1,PCL
1030 FOR PL=0 TO PCL
1040 INPUT#1,PICX(PL):INPUT#1,PICY(PL)
1050 NEXT:PICX(PCL)=9999:CLOSE#1
1060 CLS:LINE(0,0)-(639,175),PSET,7,B
1070 IF PICX(PCO)=9999 THEN RETURN
```

page 3　[File Name : SKETCH]

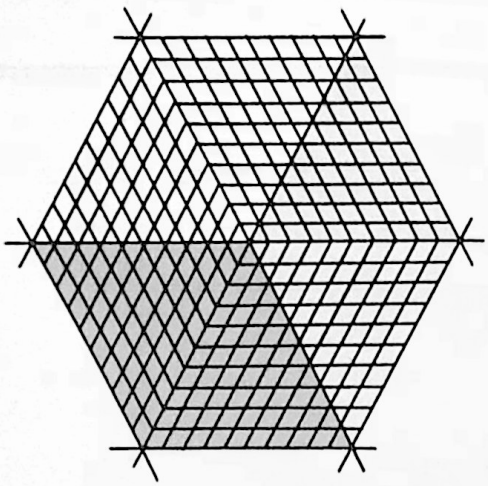
```
1080 IF PICX(PCO)=1000 THEN C=PICY(PCO):GOSUB 1140
1090 IF PICX(PCO)=2000 THEN C=PICY(PCO):GOSUB 1190
1100 IF PICX(PCO)=3000 THEN C=PICY(PCO):GOSUB 1250
1110 IF PICX(PCO)=4000 THEN C=PICY(PCO):GOSUB 1330
1120 PCO=PCO+1:GOTO 1070
1130 '---------------- LINE --------------------
1140 PCO=PCO+1:IF PICX(PCO+1)>=1000 THEN 1180
1150 X1=PICX(PCO):Y1=PICY(PCO)
1160 X2=PICX(PCO+1):Y2=PICY(PCO+1)
1170 LINE(X1,Y1)-(X2,Y2),PSET,C:GOTO 1140
1180 PSET(PICX(PCO),PICY(PCO),C):RETURN
1190 '----------------- PAINT -------------------
1200 X=PICX(PCO+1):Y=PICY(PCO+1)
1210 K(1)=PICX(PCO+2):FOR J=2 TO 7
1220 K(J)=VAL(MID$(STR$(PICY(PCO+2)),J,1)):NEXT J
1230 PAINT(X,Y),C,K(1),K(2),K(3),K(4),K(5),K(6),K(7)
1240 PCO=PCO+2:RETURN
1250 '----------------- BRUSH -------------------
1260 X1=PICX(PCO+1):Y1=PICY(PCO+1)
1270 X2=PICX(PCO+2):Y2=PICY(PCO+2)
1280 FOR B=X1 TO X2 STEP SGN(X2-X1)
1290 P=ABS(Y1-Y2):FOR I=0 TO P/5
1300 IF Y1>Y2 THEN Y=Y2 ELSE Y=Y1
1310 PSET(B,RND*P+Y,C):NEXT I,B
1320 PCO=PCO+2:RETURN
1330 '----------------- CIRCLE ------------------
1340 X1=PICX(PCO+1):Y1=PICY(PCO+1)
1350 X2=PICX(PCO+2)
1360 CIRCLE(X1,Y1),X2,C
1370 PCO=PCO+2:RETURN
1380 '****************** PSAVE ********************
1390 INPUT"FILE NAME ";N$
1400 OPEN"O",#1,N$:PRINT#1,PCO
1410 FOR PL=0 TO PCO
1420 PRINT#1,PICX(PL):PRINT#1,PICY(PL)
1430 NEXT:CLOSE#1:RETURN
1440 FOR I=0 TO PCO:PRINTPICX(I),PICY(I):NEXT
9000 '***************** V-RAM COPY SUB *****************
9010 INPUT"P-SAVE=1  P-LOAD=2 ";SL
```

page 4　[File Name : SKETCH]

```
9030 DM$=SPACE$(240)
9040 IF SL=2 THEN 9150
9050 '================== S A V E ===================
9060 INPUT"SAVE/FILE NAME ";FIN$
9070 OPEN"O",#1,FIN$
9080 FOR Y=0 TO 175
9090 GET@A(0,Y)-(639,Y),E%,G
9100 POKE &H6B00,VARPTR(E%(0)) ¥ 256
9110 POKE &H6B01,VARPTR(E%(0)) MOD 256
9120 POKE &H6B02,PEEK(VARPTR(DM$)+1)
9130 POKE &H6B03,PEEK(VARPTR(DM$)+2)
9140 EXEC&H6B04:PRINT#1,DM$;:NEXT Y:CLOSE#1:RETURN
9150 '================== L O A D ===================
9160 INPUT"LOAD/FILE NAME ";FIN$
9170 OPEN"I",#1,FIN$
9180 FOR Y=0 TO 175:DM$=INPUT$(240,#1)
9190 POKE &H6B02,VARPTR(E%(0)) ¥ 256
9200 POKE &H6B03,VARPTR(E%(0)) MOD 256
9210 POKE &H6B00,PEEK(VARPTR(DM$)+1)
9220 POKE &H6B01,PEEK(VARPTR(DM$)+2)
9230 EXEC&H6B04
9240 PUT@A(0,Y)-(639,Y),E%,PSET:NEXT Y:CLOSE#1
9250 RETURN
9260 '================ MACHINE DATA ================
9270 DATA 198,240,190,107,0,16,190,107,2,166
9280 DATA 128,167,160,192,1,38,248,57
```


◆余暇利用の協力者を求む

①パソコン用のプログラムを組む仕事を職業として4年以上経験している方
②余暇利用で在宅の仕事ですが，時間に比較的余裕のある方
③ビジネス・ユーティリティ・ゲーム・マシン語・その他のいずれかのプログラミングに自信のある方
④希望者は下記事項を書面（メモ程度で可）でお知らせ下さい（応募の秘密は厳守します）
　㋑住　所・氏　名・年　齢・自宅電話番号
　㋺現在の勤務先・職　種
　㋩プログラム（またはコンピュータ）に関する職歴およびこれまでに組んだプログラム名
　㋥所有システム・あなたの使用できる言語の種類（詳細に）・協力できる大よその月間時間
⑤郵送先：113　東京都文京区本郷5－5－18　山海堂　編集部　及川　宛



4章

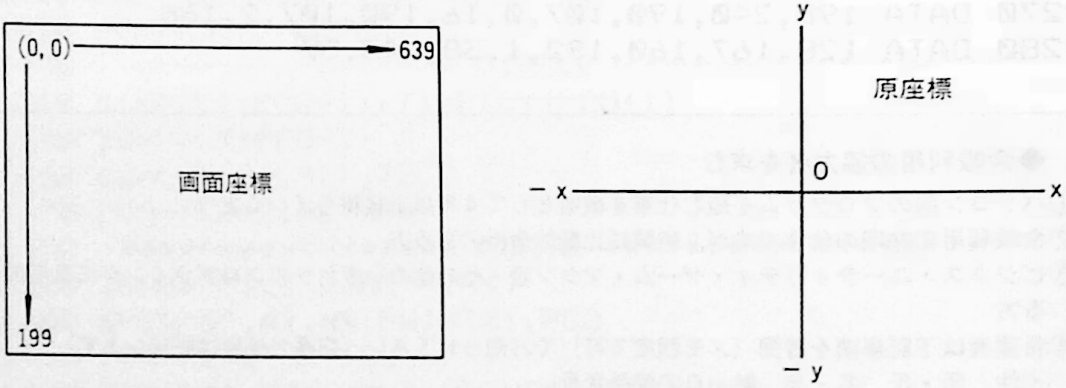
3次元グラフィックス

いよいよダイナミックな3次元グラフィックスの世界へ踏み込みます。

前章の絵を描くプログラムでは，与えられたデータに従って画面に絵を描くいわば〝静〟の図形処理でしたが，本章では，ある一連のデータを加工し，さまざまな角度や大きさの図形を描かせる〝動〟のグラフィックについて述べてみます。

画面座標系と原座標系

画面座標とは640×200ドットの座標のことをいい，原座標は数学における一般的な座標を指します。



(図4-1)

また原座標をワールド座標とも呼びますが，本書では便宜上，原座標の呼び方で統一します。

画面座標は周知の通り，垂直座標(Y)の値として画面上端が0，下端が199に固定されてお

ります。また，水平座標値(X)との画面比率も1対1になっていません。前章のいわば静的なグラフィックの実行では，この座標系でも特に支障はなかったのですが，動的な図形処理を行なう場合には，いろいろと不都合が生じてきます。一般に座標を考えるときは，X＝0，Y＝0の原点が中央にある原座標を想定するのが普通で，わかりやすいといえます。

図形の移動や拡大・縮小または回転などの処理は，架空の原座標系内で行ない，実際に描かせる時点で座標の変換を行ない，画面座標へ出力します。

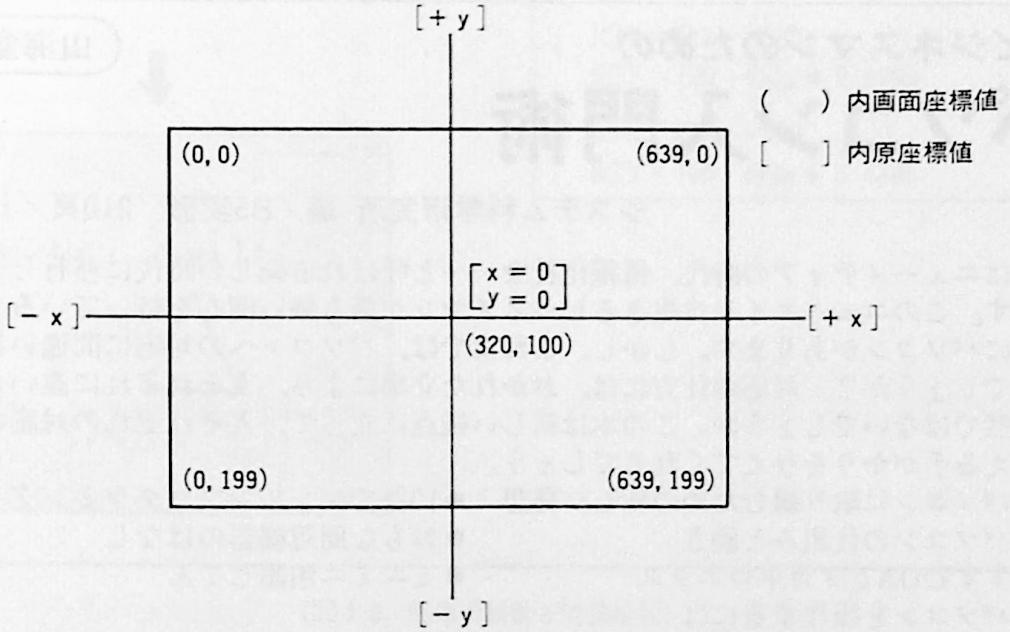
座標系の変換（画面座標←原座標）

原座標系の座標値を640×200ドットの画面座標に変換するためには，次式を用います。

画面座標 X＝原座標 x ＋320 …… ①

画面座標 Y＝100－原座標 y ×0.4495 … ②

この式によって原座標の原点（x＝0，y＝0）は，画面座標の中央（320,100）の位置にセットされます。y 座標に0.4495を掛けているのは，原座標の縦横比1対1と，画面座標の縦横比1対0.4495の調整を行なっています。この調整をしないと，描画する図形は元の図に比べ縦に伸ばされた表示になってしまいます。



（図4-2　原座標と画面座標の変換）

2次元図形の平行移動

原座標系内の図形を左右に移動させるときは，x座標に移動量を加えます。上下に動かす場合は，同様にy座標に移動量を加えます。

移動後の x＝移動前の x＋移動量　……　左右の移動

移動後の y＝移動前の y＋移動量　……　上下の移動

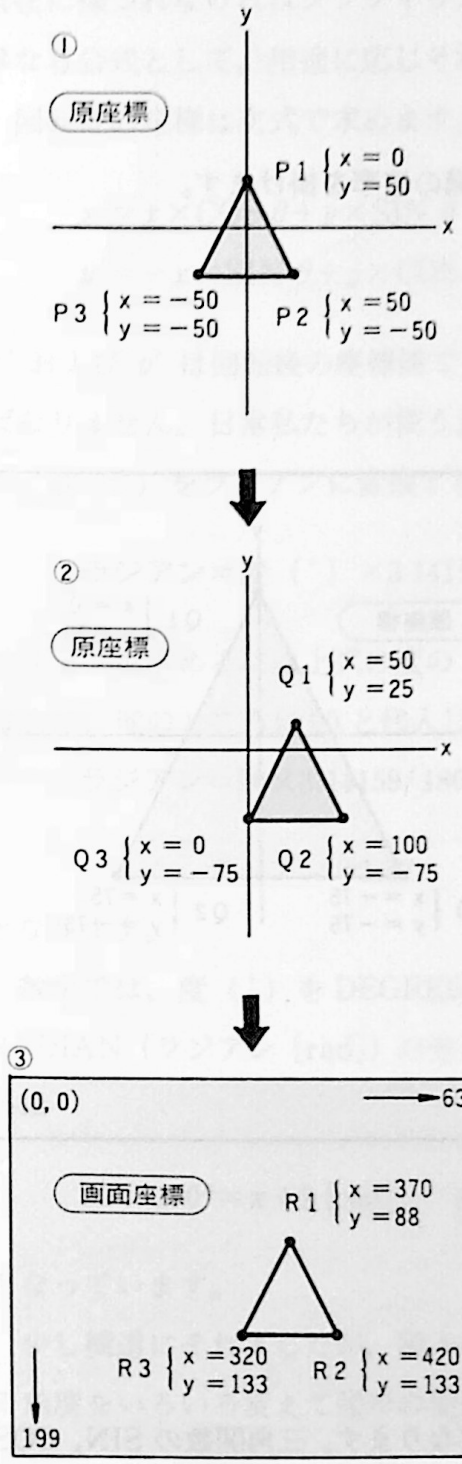
具体的な例として，原座標系内で図形を移動したのち，画面座標に変換し描画させる過程を追ってみましょう。

図4-3の①を元の図形とし右に50，下に25移動させます。P1～P3 は移動前の座標値，Q1～Q3 は移動後の座標値，R1～R3 は座標変換後の座標値です。

図4-3の処理を通して得られた R1X～R3X および R1Y～R3Y の各座標値を LINE または CONNECT で結ぶと，移動した三角形が描画できます。

```
CONNECT (R1X,R1Y)－(R2X,R2Y)－(R3X,R3Y)－(R1X,R1Y)
```

◆ビジネスマンのための
山海堂発行
パソコン入門術
システム科学研究所 編／B5変形　210頁／1500円
現代はニューメディアの時代，情報化社会……と呼ばれる新しい時代に移行しつつあります。このニューエイジに生きるビジネスマンが最も強い関心を持っているものの一つにパソコンがあります。しかし，これまでは，パソコンへの対応に間違いはなかったでしょうか？　対応の仕方には，おかれた立場により，人それぞれに違いがあって当然ではないでしょうか。この本は新しい視点に立って，人それぞれの対応の仕方を考える手がかりを与えてくれるでしょう。
●パソコンに取り組むための新しい発想
●パソコンの仕組みと働き
●すすむOAとメカトロニクス
●パソコンを操作するには
●10語でルンルンプログラミング
●おもな周辺機器のはなし
●ミニミニ用語じてん



右に 50 の移動は x に 50 を加算
下に 25 の移動は y に 25 を減算

①→②

$Q1x = P1x + 50 \quad \therefore \quad Q1x = 50$
$Q1y = P1y - 25 \quad \therefore \quad Q1y = 25$

$Q2x = P2x + 50 \quad \therefore \quad Q2x = 100$
$Q2y = P2y - 25 \quad \therefore \quad Q2y = -75$

$Q3x = P3x + 50 \quad \therefore \quad Q3x = 0$
$Q3y = P3y - 25 \quad \therefore \quad Q3y = -75$

②→③

$R1X = Q1x + 320$
$R1Y = 100 - Q1y * 0.4495$

$R2X = Q2x + 320$
$R2Y = 100 - Q2y * 0.4495$

$R3X = Q3x + 320$
$R3Y = 100 - Q3y * 0.4495$

(図 4-3　座標の移動と変換過程)

2次元図形の拡大および縮小

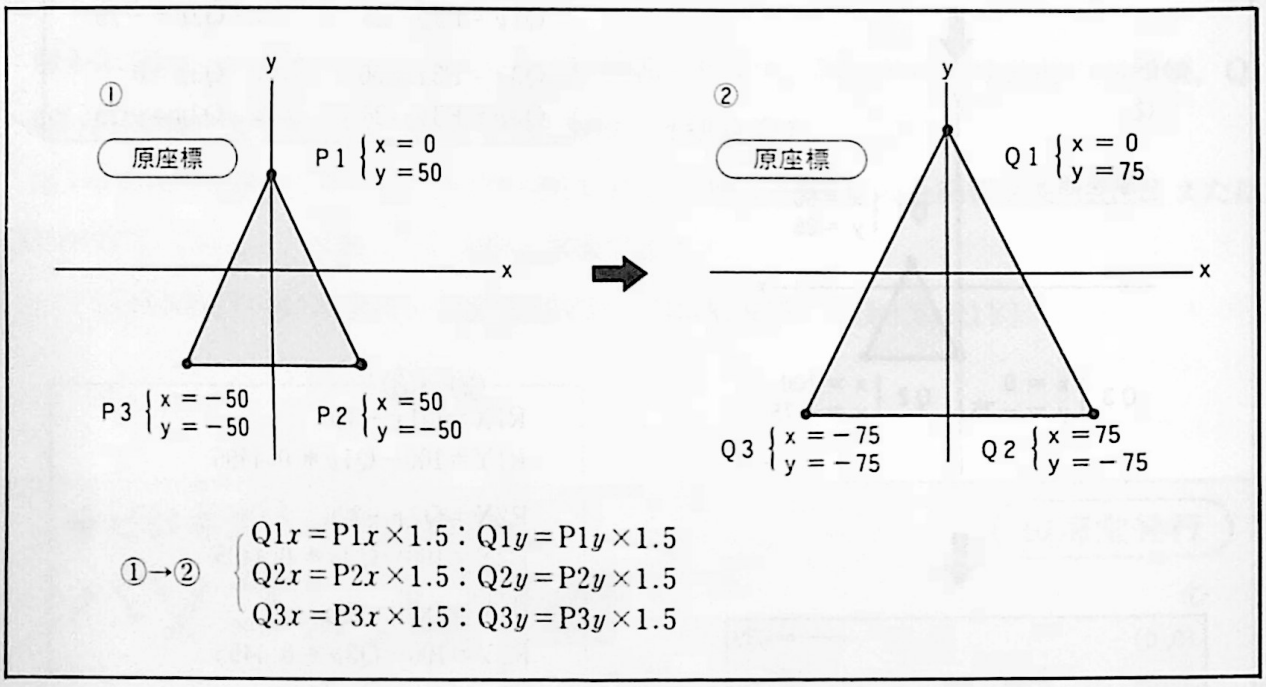
拡大はxとyに1以上の倍率を掛け，縮小は0以上1未満の倍率を掛けます。

拡大または縮小後の座標値をx'，y'とすると，

$$x' = x \times 倍率$$

$$y' = y \times 倍率$$

となります。図4-4は拡大率を1.5にしたときの例です。



(図4-4　図形の拡大)

図形の回転

回転は，これまでの移動や拡大と異なり少しめんどうになります。三角関数のSIN，COSが登場してきますので，数学アレルギーがでてくる方もいるかもしれません。しかし，SIN，COSは高度なグラフィックを実行するうえでとても便利な関数です。知識があいまいなときは，三角関数の概念だけでもおさらいをしておくと心強くなると思います。また，SIN，COSが自由

自在に操つれなければグラフィックができないと早合点されても困ります。本書で述べる式を単なる公式として，用途に応じそれぞれ使い分けてください。

回転後の座標は次式で求めます。

$$x' = x \times \mathrm{COS}\ \theta + y \times \mathrm{SIN}\ \theta$$
$$y' = -x \times \mathrm{SIN}\ \theta + y \times \mathrm{COS}\ \theta$$

x' および y' は回転後の座標値です。θ は回転角度を与えますが，ラジアンの値を入れなければなりません。日常私たちが使う角度の単位は度（°）が普通で，ラジアンはなじみがありません。度（°）をラジアンに変換する式を挙げておきましょう。

ラジアン＝度（°）×3.14159÷180

ラジアンを求めるには上式の度のところに角度を代入します。たとえば，90°をラジアンに直す場合は，度のところに 90 と代入して

ラジアン＝90×3.14159/180

∴　ラジアン≒1.57

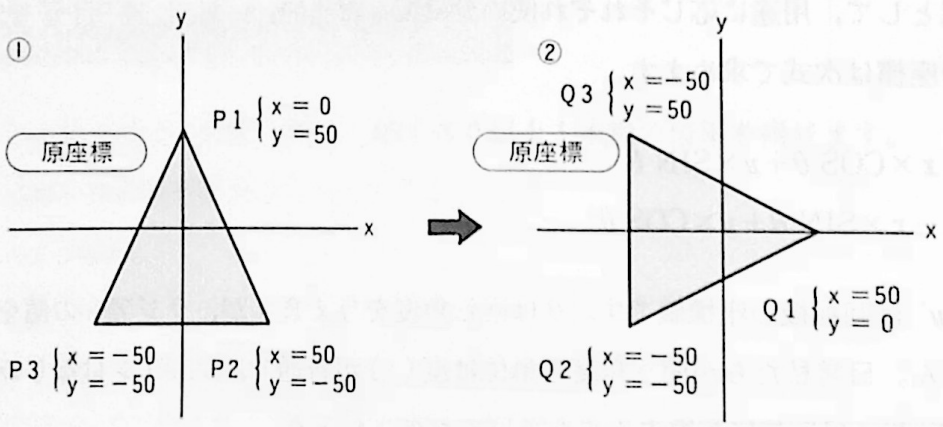
となります。

数学では，度（°）を DEGREE（ディグリー）ともいいます。また角度を表わす単位には RADIAN（ラジアン［rad］）の他に GRAD（グラード［g］）というのもあります。これらの関係は

$$90^\circ = 100^g = \pi \div 2\ [\mathrm{rad}]$$

となっています。

少し横道にそれましたが，図 4-5 に図形を 90° 回転させた実例を載せておきます。回転させる角度をいろいろ変えて図形の変化を確認してください。



$$
①\rightarrow②\begin{cases}
Q1x = P1x * \mathrm{COS}(1.57) + P1y * \mathrm{SIN}(1.57) & \therefore\ Q1x \fallingdotseq 50 \\
Q1y = -P1x * \mathrm{SIN}(1.57) + P1y * \mathrm{COS}(1.57) & \therefore\ Q1y \fallingdotseq 0 \\
Q2x = P2x * \mathrm{COS}(1.57) + P2y * \mathrm{SIN}(1.57) & \therefore\ Q2x \fallingdotseq -50 \\
Q2y = -P2x * \mathrm{SIN}(1.57) + P2y * \mathrm{COS}(1.57) & \therefore\ Q2y \fallingdotseq -50 \\
Q3x = P3x * \mathrm{COS}(1.57) + P3y * \mathrm{SIN}(1.57) & \therefore\ Q3x \fallingdotseq -50 \\
Q3y = -P3x * \mathrm{SIN}(1.57) + P3y * \mathrm{COS}(1.57) & \therefore\ Q3y \fallingdotseq 50
\end{cases}
$$

(図 4-5　図形の回転)

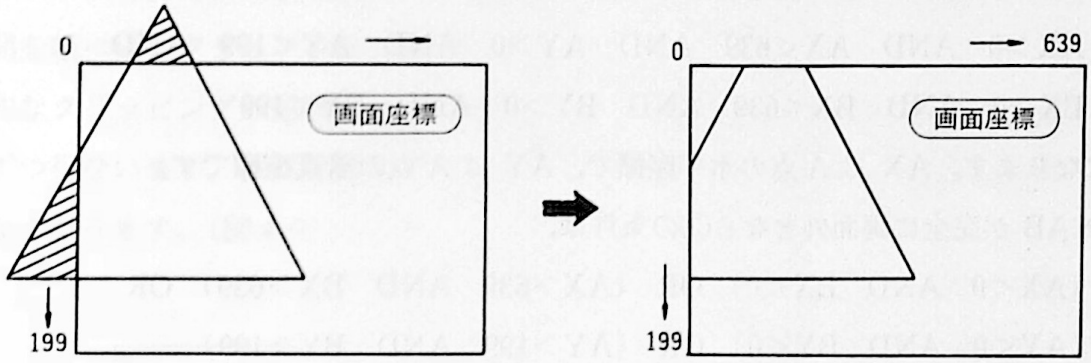
大切なクリッピング

原座標系での拡大・縮小・回転は前節の図 4-4 および 4-5 で実例を述べましたが，画面に出力するときは図 4-3 と同様に画面座標に変換してから描画させます。変換した座標の値が

$0<X<639$,　$0<Y<199$　の範囲ならば問題はないのですが，拡大に伴ってこの範囲を超えてしまうとどうなるでしょうか。

F−BASIC　Ver.3 では，LINE 文などで線を引く場合，X と Y の値が負になるとエラーになり，$X>640$, $Y>200$ の時は正しい表示になりません。

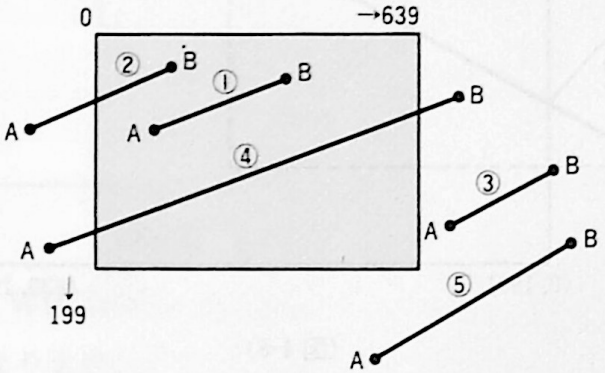
クリッピングとは，画面から溢れてしまった線を適切なところで切る働きをいいます。



(図4-6　クリッピング)

クリッピングの条件

画面を横切る線は無数に存在しますが，図4-7を例に線分ABのクリップについて考えてみましょう。



(図4-7　クリッピングの条件)

(1)　①の線分は，A点B点とも画面内ですので，クリッピングの処理は必要なく，そのまま描画できます。

(2)　②の線分はA点が画面外なのでクリップしなければなりません。

(3)　③の線分はA点B点両方のX値が640を超えていますので描画の必要はありません。

(4)　④および⑤の線分はA点B点とも画面外ですが，画面を横切る場合とそうでない場合があります。

線分 AB が画面内の①を満足させる条件は，

AX＞0　AND　AX＜639　AND　AY＞0　AND　AY＜199　AND

BX＞0　AND　BX＜639　AND　BY＞0　AND　BY＜199

が真になります。AX は A 点の水平座標で，AY は A 点の垂直座標です。

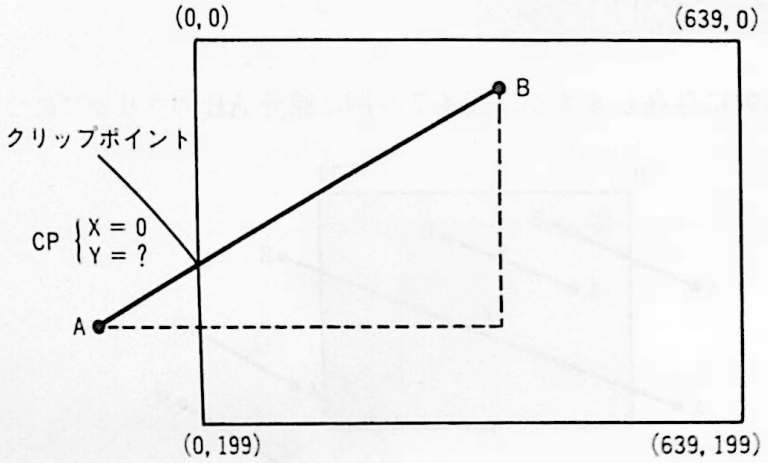
線分 AB が完全に画面外となる③の条件は，

(AX＜0　AND　BX＜0) OR (AX＞639　AND　BX＞639) OR

(AY＜0　AND　BY＜0) OR (AY＞199　AND　BY＞199)

を満足します。

さて，前記2つの条件式の結果が，虚であるならば②④⑤のどれかにあてはまり，クリッピングの処理をしなければなりません。図4-8の線分 AB を例にとり説明します。



(図4-8)

クリップポイント CP の座標がわかれば，画面内の B 点と CP を結び，クリップされた線を引くことができます。CP の X 値は画面左端であることから，0であることがわかります。不明なのは CP の Y 値ですが，これは三角形の比で求めることが可能です。

(BX－AX)：(AY－BY)＝(CPX－AX)：(AY－CPY)

以上の比例式が成り立ちますので，

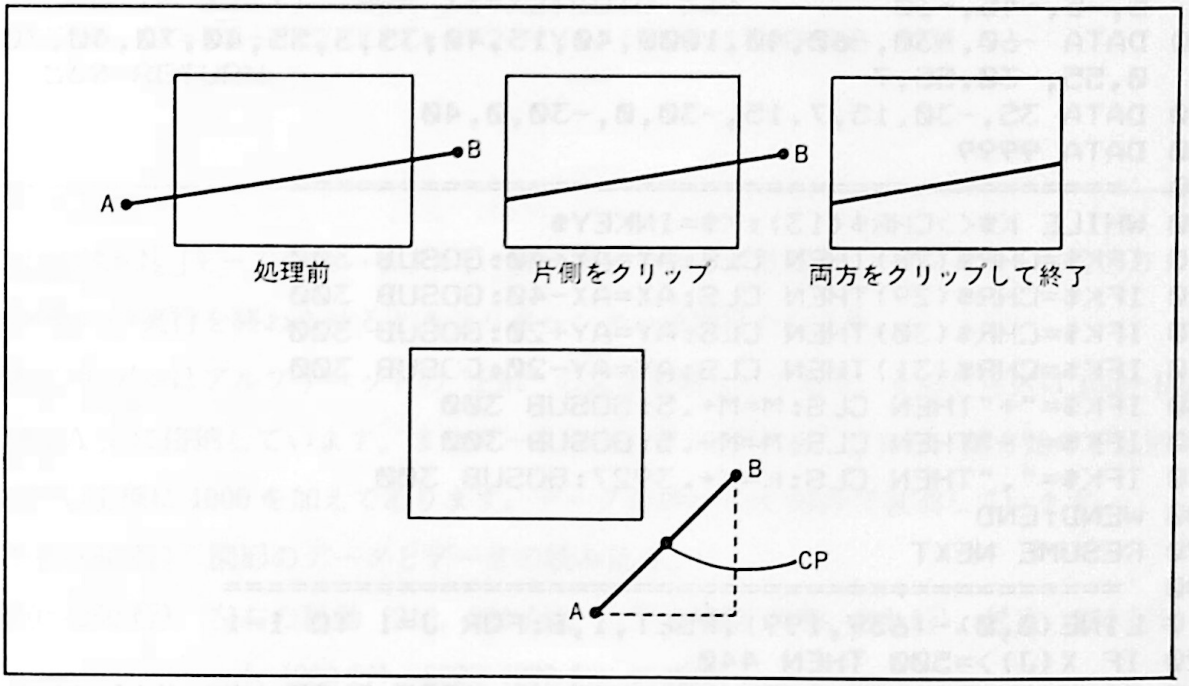
$$CPY = AY - \frac{(AY-BY)\times(0-AX)}{BX-AX}$$

この式により CPY が導きだせます。CPX に0を代入し，

LINE (BX, BY)−(CPX, CPY), PSET

と2点間を結べば，クリップされた線を引くことができます。

基本的なクリッピングの処理は以上ですが，図4-7の④のように両端にまたがる線分の場合は片側ずつ行ないます。また⑤のときは，CPY が画面外（CPY＞200）になりますので描画の必要はなくなります。（図4-9）



（図4-9　クリッピングの手順）

F-BASIC に VIEW や WINDOW の命令があれば，もっと簡単な処理ですむのですが，無いものねだりをしても始まりません。

BASIC の命令が無いのであれば，同等の機能を有するルーチンを作ってしまうのもプログラミングの楽しみといえます。これまでに述べた処理法に手を加えると，VIEW や WINDOW のサブルーチンができますので，力のある方は挑戦してみてください。

図形加工のサンプルプログラム

リスト4-1は図形の移動・拡大・縮小・回転・クリッピングを実行するサンプルプログラムです。カーソルキーを押すと図形は矢印の方向に平行移動します。[+]キーは拡大、[−]キーは縮

(リスト4-1)

```
100 '■ ｽﾞｹｲ ﾉ ｲﾄﾞｳ･ｶｸﾀﾞｲ･ｼｭｸｼｮｳ･ｶｲﾃﾝ * (C) S.Tajima ■
110 '
120 DIM X(100),Y(100):CLS:M=1:I=1:ON ERROR GOTO 290
130 READ X:IF X=9999 THEN C=4:GOSUB 300:GOTO 200
140 READ Y:X(I)=X:Y(I)=Y:I=I+1:GOTO 130
150 DATA 940,40,-10,40,-10,25,-45,25,-45,10,-20,10,-20,-5,-4
    5,-5,-45,-30
160 DATA -60,-30,-60,40,1000,40,15,40,35,3,55,40,70,40,70,-3
    0,55,-30,55,7
170 DATA 35,-30,15,7,15,-30,0,-30,0,40
180 DATA 9999
190 '=========================================
200 WHILE K$<>CHR$(13):K$=INKEY$
210 IFK$=CHR$(28)THEN CLS:AX=AX+40:GOSUB 300
220 IFK$=CHR$(29)THEN CLS:AX=AX-40:GOSUB 300
230 IFK$=CHR$(30)THEN CLS:AY=AY+20:GOSUB 300
240 IFK$=CHR$(31)THEN CLS:AY=AY-20:GOSUB 300
250 IFK$="+"THEN CLS:M=M+.5:GOSUB 300
260 IFK$="-"THEN CLS:M=M-.5:GOSUB 300
270 IFK$=","THEN CLS:K=K+.3927:GOSUB 300
280 WEND:END
290 RESUME NEXT
300 '=========================================
310 LINE(0,0)-(639,199),PSET,1,B:FOR J=1 TO I-1
320 IF X(J)>=500 THEN 440
330 IF X(J-1)>=500 THEN LX1=X(J-1)-1000 ELSE LX1=X(J-1)
340 LX2=X(J)
350 X1=LX1*COS(K)+Y(J-1)*SIN(K):Y1=-LX1*SIN(K)+Y(J-1)*COS(K)
360 X2=X(J)*COS(K)+Y(J)*SIN(K):Y2=-X(J)*SIN(K)+Y(J)*COS(K)
370 X1=X1*M+AX+320:Y1=100-(Y1*M+AY)*.4495
380 X2=X2*M+AX+320:Y2=100-(Y2*M+AY)*.4495
390 '=========================================
400 IF X1>0 AND X1<638 AND Y1>0 AND Y1<198 AND X2>0 AND X2<6
    38 AND Y2>0 AND Y2<198 THEN 430
410 IF (X1<0 AND X2<0) OR (X1>638 AND X2>638) OR (Y1<0 AND Y
    2<0) OR (Y1>198 AND Y2>198) THEN 440
420 GOSUB 460:SWAP X1,X2:SWAP Y1,Y2:GOSUB 460
430 LINE(X1,Y1)-(X2,Y2),PSET,C
440 NEXT:RETURN
450 '=========================================
460 IF X2<1 THEN XX=1:GOTO 490
470 IF X2>638 THEN XX=638:GOTO 490
```

```
480 XX=X2
490 IF X2=X1 THEN YY=Y2:GOTO 510
500 YY=Y2-(Y2-Y1)*(X2-XX)/(X2-X1)
510 IF YY<1 THEN YY=1:GOTO 540
520 IF YY>198 THEN YY=198:GOTO 540
530 X2=XX:Y2=YY:GOTO 560
540 IF Y2=Y1 THEN XX=X2:GOTO 530
550 XX=X2-(X2-X1)*(Y2-YY)/(Y2-Y1):GOTO 530
560 RETURN
```

小です。回転は[,]キーで行ないますが，一度に回転できる角度は22.5度にセットしてあります。プログラムの実行を終わらせるときはリターンキーを空打ちします。

図形のモデルはアルファベットの"FM"の文字を形どっており，そのデータは150～180行のDATA文に格納しています。またデータはX，Yの順で並んでおり，描き始めを区別するためにX座標に1000を加えてあります。データの終わりは9999で区別しています。

(100～180行)　図形のデータとデータの読み込み。

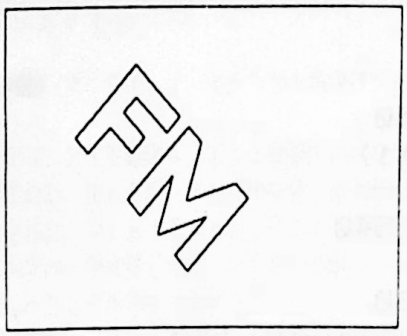
(200～280行)　左右の移動（210，220行），上下の移動（230，240行），拡大（250行），縮小（260行），回転（270行）の選択。

(310～380行)　回転と座標決定ルーチン。

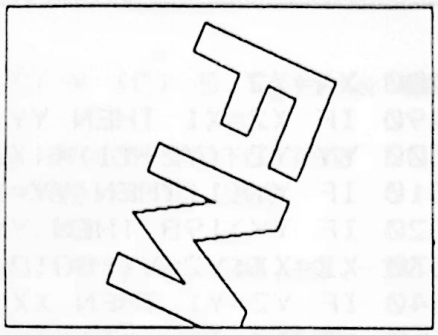
(400～560行)　クリッピングと描画のサブルーチン。

このプログラムで特に重要なところは370行と380行です。X1，Y1，X2，Y2はクリップ前の決定座標値ですが，図形の加工や原座標から画面座標の変換を行なっています。変数Mは拡大率，AXは左右の移動量，AYは上下の移動量です。この行で決まった座標値を次行からのクリッピングルーチンに通し，430行のLINE文で描画させています。

リスト4-2はカセットハーフの図形データです。リスト4-1のDATA文をこのリストの内容に打ち変えて図形の変化を楽しんでください。



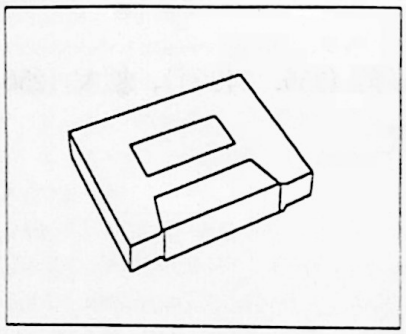
(図4-10)



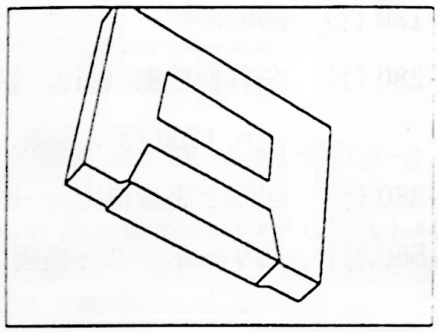
(図4-11)

(リスト4-2)

```
150 DATA 942,-55,-108,24,-106,8,-58,-72,-32,-62,-30,-64,66,-
    29,69,-25,88,-18
155 DATA 90,-2,68,-10,64,-10,-32,-43,-35,-47,-58,-55,-58,-72
160 DATA 1090,-2,32,64,-108,24,942,16,-44,-6,28,17,12,36,-58
    ,16
170 DATA 968,-43,-38,-20,40,6,64,-10
175 DATA 970,-42,-30,-64,1064,-10,63,-30
180 DATA 9999
```



(図4-12)



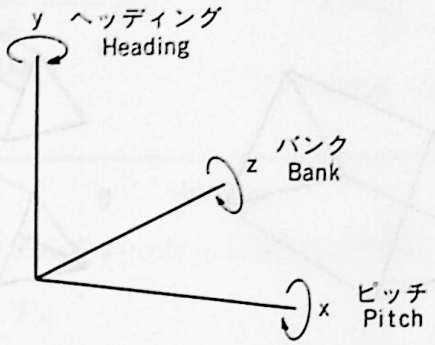
(図4-13)

3次元座標

3次元を表わす座標は，これまでの2次元座標 x，y にもう一本 z を加え，x，y，z で構成されます。移動や拡大は2次元の場合と同じ処理で可能ですが，回転となると少しめんどうになってきます。2次元では回転軸が一本ですんだものが，3次元では無数の回転軸が存在します。しかし，無数ではあまりにもとりとめがないので，普通は x 軸，y 軸，z 軸の3本を対象

にしています。いい方を変えると3次元の形状はx軸，y軸，z軸の3方向の回転の組み合わせで全容を見ることができます。3次元グラフィックスの醍醐味は，これらの各方向の回転を通して，あらゆる角度からモデルの形状を解析することにあるわけです。

一般に，x軸の回転をピッチ（またはチルト），y軸の回転をヘッディング（またはスピン），z軸の回転をバンクと呼んでいます。



（図4-14　3次元座標）

3次元モデルの形状作成手法

3次元モデルの図形表現には，

(1)　ワイヤーフレーム（針金細工風）

(2)　サーフェィスモデル（色紙細工風）

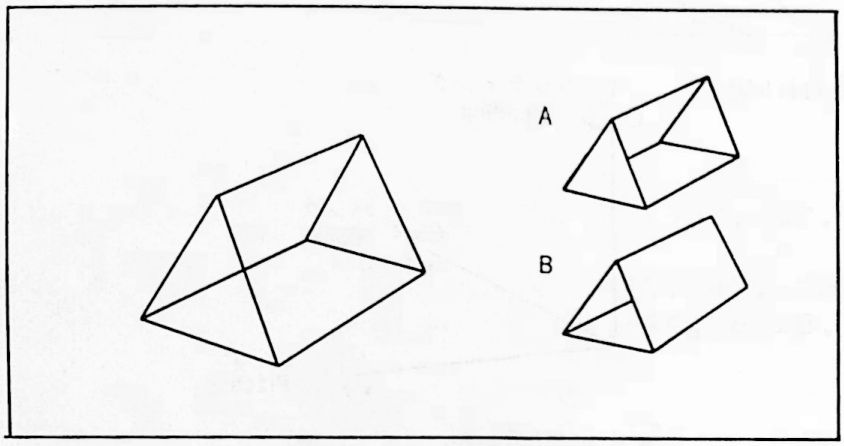
(3)　ソリッドモデル（積木細工風）

の3方式が，CADの分野では，それぞれ用途に応じて用いられています。

CAD（キャド）とは，Computer　Aided　Designの略で〝コンピュータによる支援設計〟と訳されます。もともと飛行機，造船関係で普及したCADですが，最近ではハードのコストパフォーマンスが高まるとともに，さまざまな分野で使われています。

パソコンで3次元図形の表現を行なうときは，(1)のワイヤーフレームの手法を使うのが一般的です。サーフェィスやソリッドもできないわけでもありませんが，かなり限定された表現になってしまうでしょう。あたかも，軽自動車に何十トンものトレーラーを引かせるようなもので，パソコンレベルのハードでは無理が生じてきます。

ワイヤーフレームは，比較的簡単なアルゴリズムで描くことができるのですが，反面，形状の構成がわかりにくいという短所もあります。図4-15の左図は，三角柱をワイヤーで表現したものですが，AにもBにも見えてしまいます。向こう側の線をある所で切ってしまう陰線の処理が，完璧に計算できると良いのですが，複雑な形状を持ったワイヤーの陰線消去は大変難しくなります。



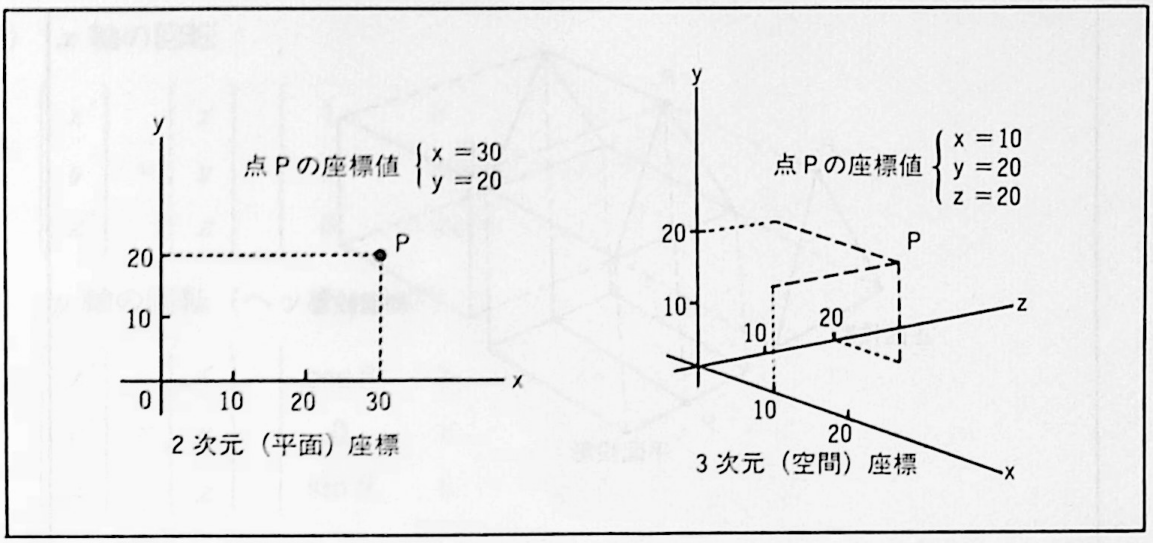
(図4-15)

座標値と3面図との対応

2次元図形では比較的簡単だった座標値の読み取りも，3次元のzが加わった空間座標になると，とたんに混乱しそうになってきます。2次元のときのように，グラフ用紙に絵を描き，グラフの目盛からxとyを読むという単純な作業の延長では，3次元座標の値をプロットすることは，複雑なモデルほど不可能になってきます。図4-16に点Pとその座標の関係を示してみました。見てわかるように3次元の空間座標における座標値の読み取りは，とても複雑です。よほどの天才でない限り，頭の中でこの空間を想定し，何十何百もの点を間違いなくプロットすることはできません。

せっかく3次元グラフィックスをやるのですから，飛行機や車などのデータを入れて，さまざまな方向からの図を描かせてみたいとは思っても，肝心のデータ入力でつまずいては何にもなりません。

そこで，三面図の登場です。三面図は建築はもとより，あらゆる産業の分野で使われており，正面，側面，平面の各方向から見た図を1枚の紙にまとめた図面であることは周知の通りです。



(図 4-16)

3次元モデルの各座標値は，モデルをいったん三面図に描き起こしたうえ，その図を見ながらプロットすると大変楽になります。

図 4-17 は三角柱のモデルとその三面図です。三角柱の点 P は，三面図の P′ に対応しています。図 4-16 と比べ，座標の読み取りが楽で，そのうえ正確な値を入力できます。

CAD など実務的な 3D.C.G（3次元コンピュータグラフィックス）のプログラムでは速く，かつ，わかり易いモデルの形状をいかに表現するかという点も重要視されていますが，それよりも，いかにモデルのデータが入力し易く，また，データの訂正がやりやすいかのほうが課題となっています。データベースができてしまえば，そのデータを加工し料理することはコンピュータの得意とするところですが，基礎的なデータ作りは，人間の手に負うところが多いのが現状です。

◆実務に役立つ (FM-7・NEW7・77)

FM ディスクファイルの活用法（仮題）

(60年 2 月刊)

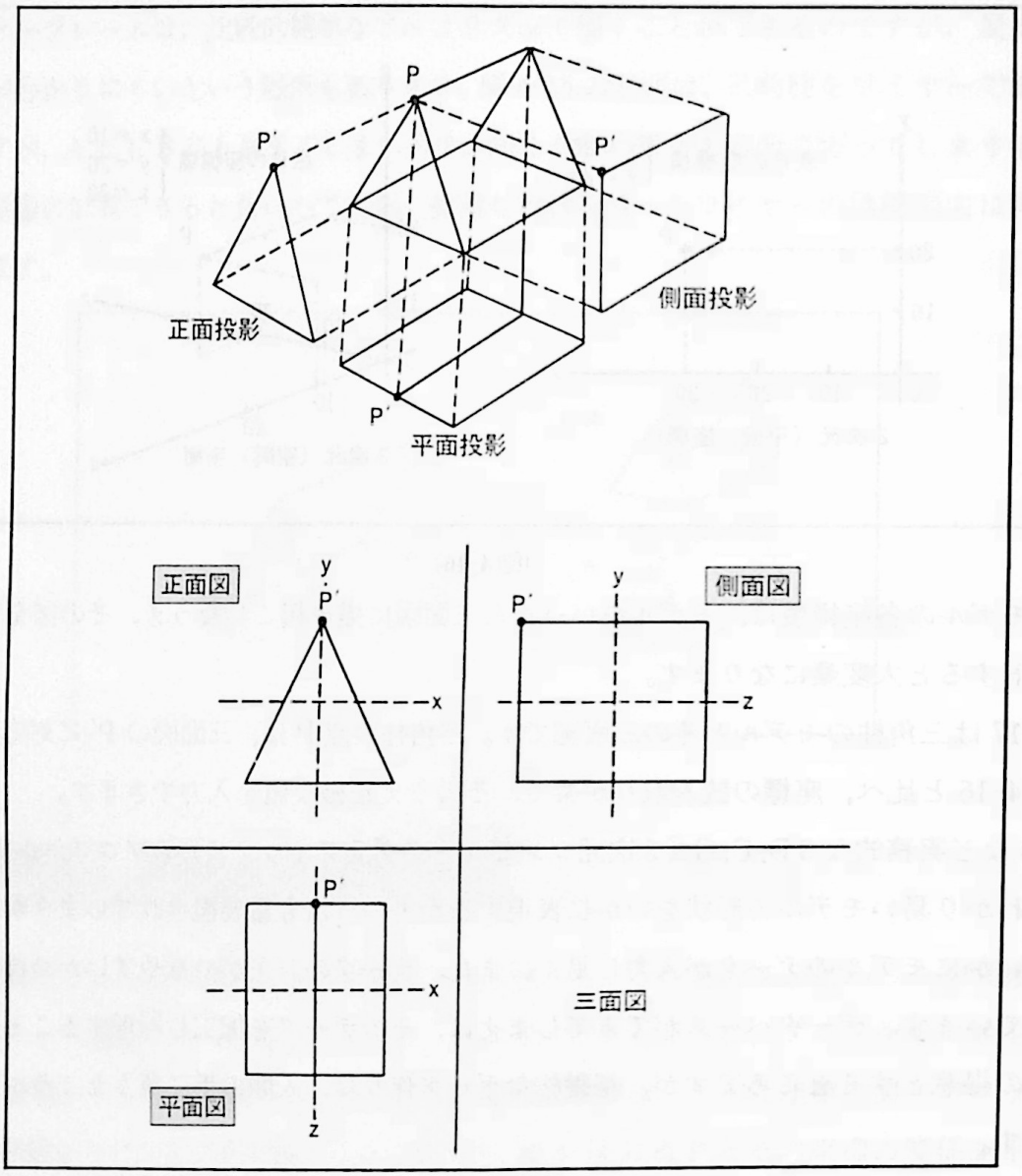
ディスクを使うことによってパソコン活用の範囲は大きく広がります．実務に役立つ使い方の基礎から応用までを入門者にも分かるようていねいに解説しています．

◆BASIC から入る (FM-7・NEW7・77)

FM マシン語入門の手引き （仮題）

(60年 2 月刊)

興味津々のマシン語を読者のなじんでいる BASIC からアタック．初めてマシン語をかじってみようと思っている人，足踏みしている人へのおすすめ本です．



(図 4-17)

3 次元空間の座標変換

x, y, z の各座標軸を回転させ，回転後の新しい座標を導きだすためには次の式を用います。x', y', z' は回転後の座標値，x, y, z は回転前の座標値です。θ は回転角度でラジアンが単位です。

(1) x 軸の回転

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix}$$

(2) y 軸の回転（ヘッディング）

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix}$$

(3) z 軸の回転

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

あまり，なじみのない式の形ですが，これは「行列」matrix で次のように計算します。

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} = \begin{pmatrix} m_{11}x + m_{12}y + m_{13}z \\ m_{21}x + m_{22}y + m_{23}z \\ m_{31}x + m_{32}y + m_{33}z \end{pmatrix}$$

「行列はどうも苦手だ」という意識が頭をもたげてくる方もいるかもしれません。以前は行列は難しすぎると，高校のカリキュラムに入ってなかったぐらいですから当然のことです。

もう少しわかりやすく，普通の式に直してみましょう。

(1) x 軸の回転

$x' = x$

$y' = y \times \cos\theta + z \times \sin\theta$

$z' = -y \times \sin\theta + z \times \cos\theta$

(2)　y 軸の回転

$$x' = x \times \cos\theta - z \times \sin\theta$$

$$y' = y$$

$$z' = x \times \sin\theta + z \times \cos\theta$$

(3)　z 軸の回転

$$x' = x \times \cos\theta + y \times \sin\theta$$

$$y' = -x \times \sin\theta + y \times \cos\theta$$

$$z' = z$$

これでだいぶ身近な形の式になりましたが，式だけでは，まだピンときませんので，実例を挙げておきます。

$x=100$，$y=100$，$z=0$ の座標値を持つ点 P を想定して，x 軸を中心に 45° おきに 180° 回してみます。リスト 4-3 は，この実行プログラムと実行結果の印字リストです。見ていただければわかると思いますが，一応リストの説明をしておきます。

(リスト 4-3)

```
100 X=100:Y=100:Z=0
110 FOR D=0 TO 180 STEP 45
120  R=D*3.14159/180
130  XX=X
140  YY=Y*COS(R)+Z*SIN(R)
150  ZZ=-Y*SIN(R)+Z*SIN(R)
160  PRINT D,FIX(XX),FIX(YY),FIX(ZZ)
170 NEXT D

Ready
RUN
 0              100            100             0
 45             100            70             -70
 90             100            0              -100
 135            100           -70             -70
 180            100           -100             0

Ready
HARDC
```

(100行)　x，y，z の座標値をセット。

(110行)　0°から180°まで45°おきに回転を設定。

(120行)　設定した回転角度Dをラジアンに変換。

(130～150行)　x 軸まわりの回転式。(1)式と同じです。XX，YY，ZZ に新座標がセットされる。

(160行)　x，y，z の順で，回転後の新座標を印字する。

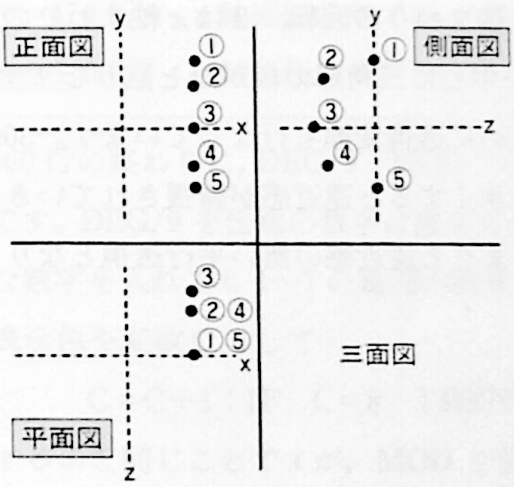
(170行)　ループ。

さて，このプログラムにより，x 軸まわりに45°ずつ回転した新しい座標値がわかりました。プリントされた数字だけではまだ分かりづらいので，x，y，z の各座標値を三面図にプロットしてみます。

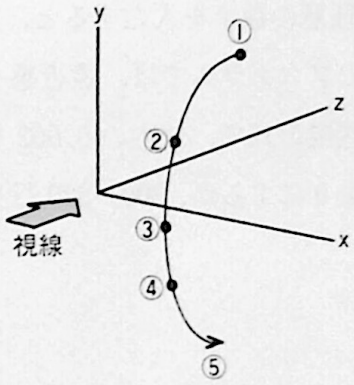
図4-18が各座標点を拾った図で，①が0°，②が45°，③が90°，……となっています。角度によって点Pの座標値が変わり，動きがはっきりわかると思います。この動きを三次元的に視覚化したのが図4-19です。

リスト4-3の140～150行を(2)式または(3)式に入れ替えて y 軸まわり，z 軸まわりの回転も試してください。

x'，y'，z' はそれぞれ XX，YY，ZZ に，θ は (R) に変更してください。



(図4-18)

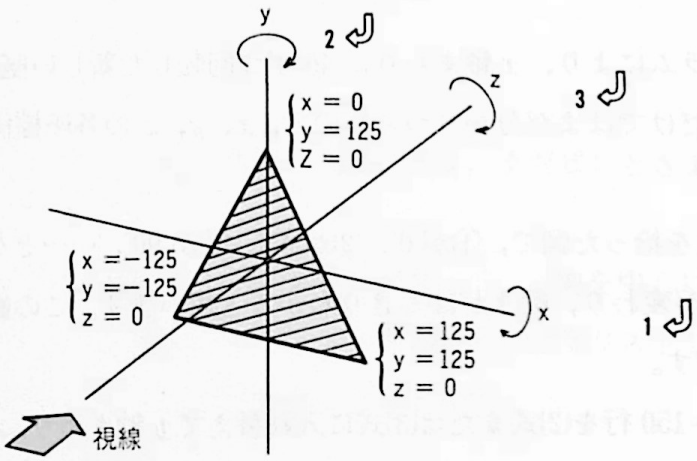


(図4-19)

各軸まわりの回転実験1

リスト4-3をもう少し発展させ，x軸，y軸，z軸ごとの回転を実際に目で確かめるプログラムとして，リスト4-4を用意しました。

回転させるモデルは図4-20で示す三角形の板です。



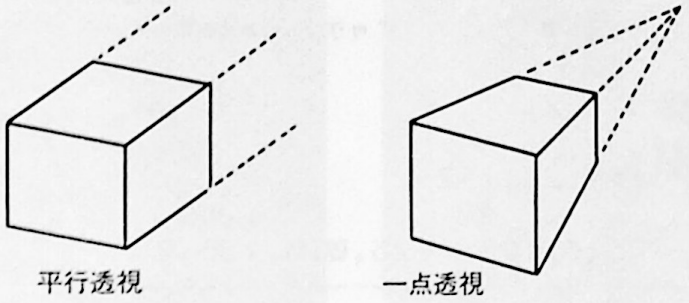
(図4-20)

RUN ↵とキーインすると，

1 OR 2 OR 3?

と聞いてきます。1はx軸まわりの回転，2はy軸まわりの回転，3はz軸まわりの回転です。1〜3の任意の数字を入力すると，指定した軸を中心に三角形の板がひと回りします。

なお，このプログラムでは，遠近感を表現するために透視変換も行なっています。300〜320行が1点の透視の式で，式内の0.002を少しずつ大きくすると遠近感が誇張されていきます。また0.002を0にするか，300〜320行を削除してしまうと遠近感の無い平行透視となります。



(図 4-21　透視変換)

(100～160 行)　三角形の各頂点の座標をデータ文から読み取り，X(n)，Y(n)，Z(n) の配列変数に格納。

(190～230 行)　メニューの表示と選択。

(250 行)　9°おきに，0°から 360°の回転を設定。

(260 行)　度（°）からラジアンの変換。

(270～280 行)　座標計算の準備。

(290 行)　メニューで選んだ数字に従ってサブルーチンに飛ぶ。

(300～320 行)　透視変換。

(330～340 行)　原座標系から画面座標への変換。

(360～380 行)　描画と消去。370 行は表示時間のタイミング。

(410～430 行)　x 軸回転のサブルーチン。

(450～470 行)　y 軸回転のサブルーチン。

(490～510 行)　z 軸回転のサブルーチン。

360 行の終わりに，DEG/9　MOD　7+1 とあるのは，線を引くたびに表示色を変えるためです。DEG/9 を任意の数字に置きかえ，ダイレクトモードでこの式を試してみると，どんな数字を入れても 1～7 の範囲の結果になります。

表示色を変数 C として

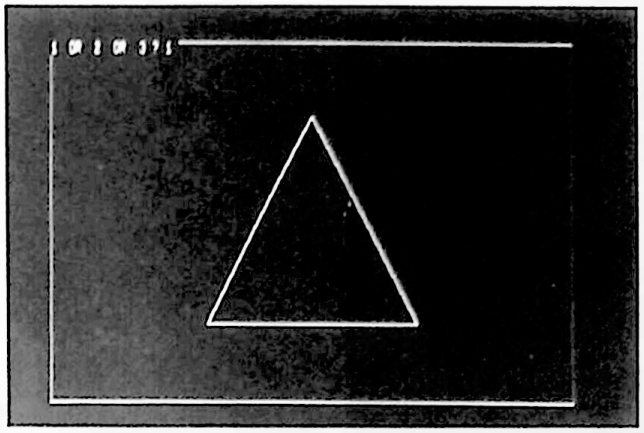
```
C=C+1 : IF  C=8  THEN  C=0
```

とするのと同じことですが，MOD を使ったほうがスマートといえます。

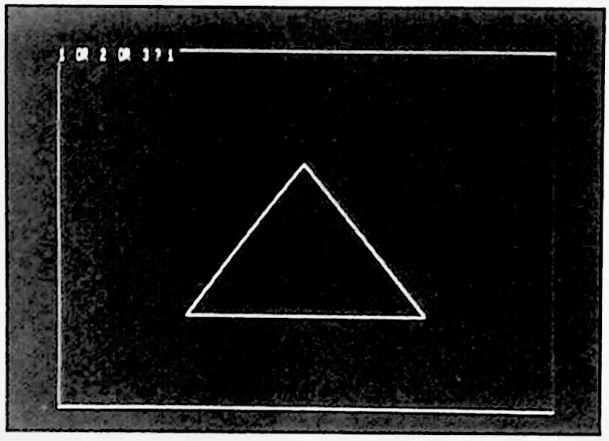
```
C=n  MOD  7+1
```

(リスト4-4)

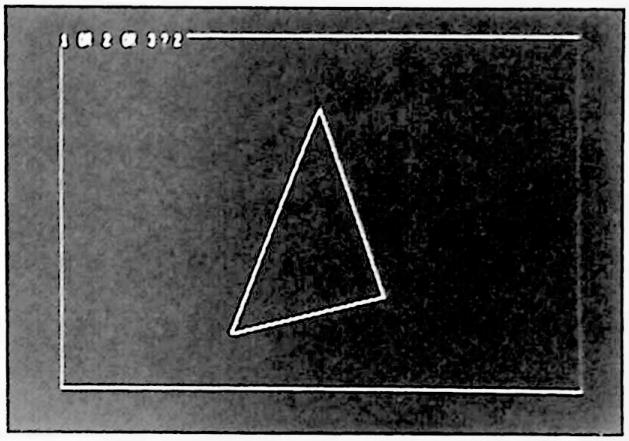
```
100 '******  X,Y,Z カイテン ノ ジッケン  ******
110 CLS
120 FOR I=1 TO 3
130  READ X,Y,Z
140  X(I)=X:Y(I)=Y:Z(I)=Z
150 NEXT I
160 DATA-125,-125,0,125,-125,0,0,125,0
180 '---------------------------------
190 LINE(0,0)-(639,199),PSET,1,B
200 LOCATE 0,0
210 INPUT"1  OR  2  OR  3 ";F
220 IF F=1 OR F=2 OR F=3 THEN GOSUB 250
230 GOTO 190
240 '---------------------------------
250 FOR DEG=0 TO 360 STEP 9
260  RAD=DEG*3.14159/180
270  FOR J=1 TO 3
280   X=X(J):Y=Y(J):Z=Z(J)
290   ON F GOSUB 410,450,490
300   PX=X1/(Z1*.002+1)
310   PY=Y1/(Z1*.002+1)
320   PZ=Z1/(Z1*.002+1)
330   XX(J)=PX+320
340   YY(J)=100-PY*.4495
350  NEXTJ
360  CONNECT(XX(1),YY(1))-(XX(2),YY(2))-(XX(3),YY(3))-(XX(1)
    ,YY(1)),DEG/9 MOD 7+1
370  FOR T=0 TO 300:NEXT T
380  CONNECT(XX(1),YY(1))-(XX(2),YY(2))-(XX(3),YY(3))-(XX(1)
    ,YY(1)),0
390 NEXT DEG:RETURN
400 '::::::::::::::::::::::::::::::::::
410 X1=X
420 Y1=Y*COS(RAD)-Z*SIN(RAD)
430 Z1=Y*SIN(RAD)+Z*COS(RAD):RETURN
440 '---------------------------------
450 X1=X*COS(RAD)-Z*SIN(RAD)
460 Y1=Y
470 Z1=X*SIN(RAD)+Z*COS(RAD):RETURN
480 '---------------------------------
490 X1=X*COS(RAD)-Y*SIN(RAD)
500 Y1=X*SIN(RAD)+Y*COS(RAD)
510 Z1=Z:RETURN
```



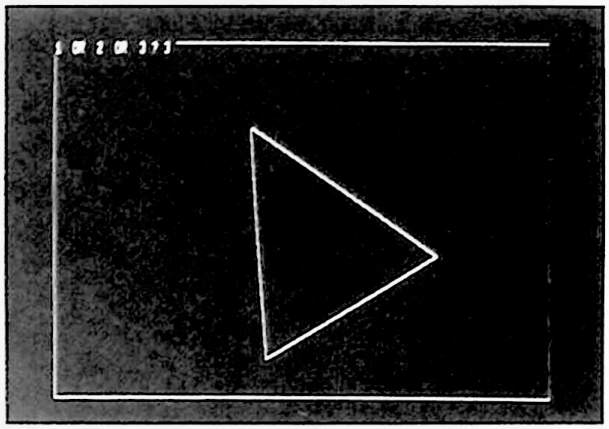
(写真 4-1　回転前の[4])



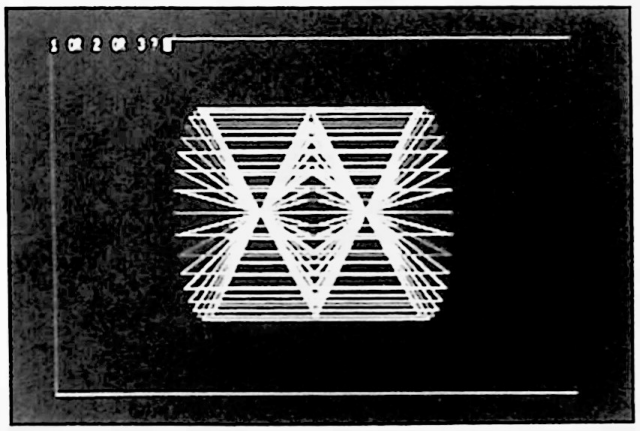
(写真 4-2　x 軸回りの回転)



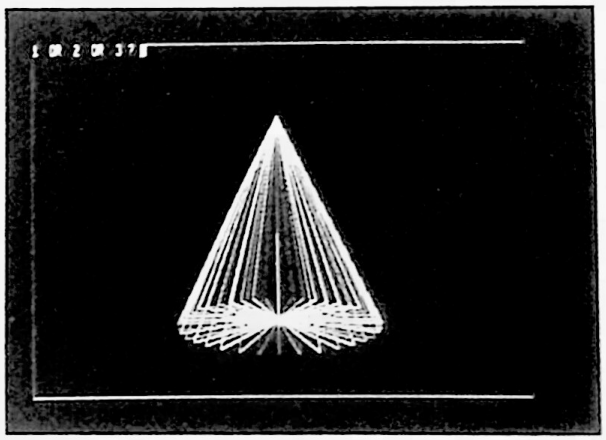
(写真 4-3　y 軸回りの回転)



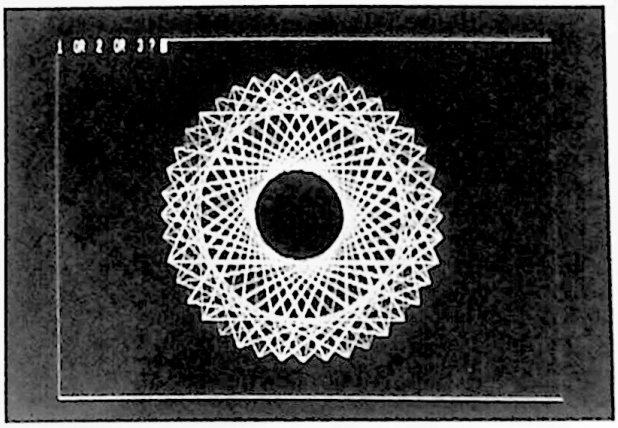
(写真 4-4　z 軸回りの回転)



(写真 4-5　x 軸回りの連続回転)



(写真 4-6　y 軸回りの連続回転)



(写真 4-7　z 軸回りの連続回転)

回転実験プログラム

前記リスト 4-4 の実験プログラム 1 は各軸ごとの回転で，たとえば x 軸まわりを 45°にして，かつ，y 軸まわりを 30°にするという組み合わせまではできません。次に示すリスト 4-5 は，3 次元モデルを観察するうえで，必要な条件をほぼ満足させるプログラムです。

機能は以下の通りです。

(1)　図形の平行移動

(2)　図形の拡大（近ずく），縮小（遠のく）

(3)　x，y，z 軸まわりの組み合わせ回転

(4)　透視変換

(5)　画面座標におけるクリッピング

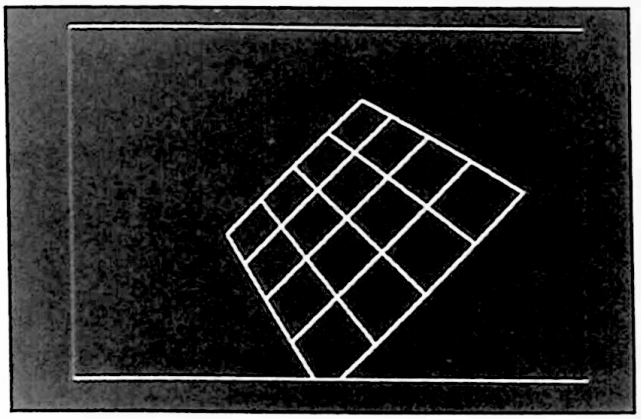
サンプルモデルとして，格子状のモデルをデータ文に格納してあります。このモデルはフライト・シュミレーターや，3D のデモなどによく使われる形状です。座標データは x，y，z の順で並んでおり，描き始めを区別するために，その場合は実際の x 座標値に 1000 を加えてあります。また，データエンドは 9999 としました。

操作法は画面に出力されませんので，次に示しておきます。

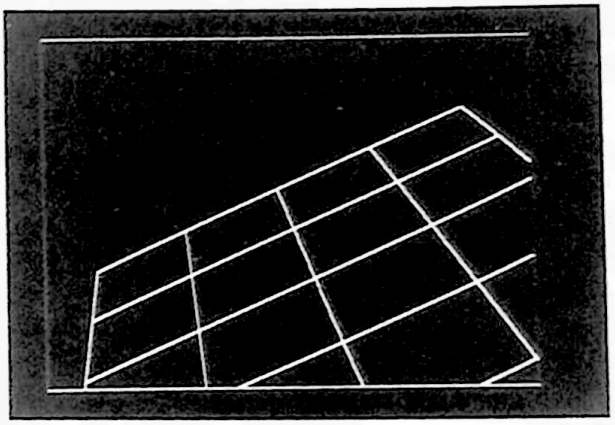
← →	図形の左右移動
↑ ↓	図形の上下移動
+	図形を1.5倍拡大
−	図形を0.5倍拡大（縮小）
2 8	ピッチング（x軸まわりの回転）
4 6	ヘッディング（y軸まわりの回転）
, =	バンク（z軸まわりの回転）
↵	終了

回転はキーを押すごとに22.5度ずつ回転表示されます。なお，このプログラム(リスト4-5)は，リスト4-1の2次元図形の回転プログラムとリスト4-4の3次元各軸ごとの回転プログラムをベースにしてあります。

(120～180行)　データの読み込み。
(200～320行)　機能の選択。
(360～430行)　描画準備と画面座標への変換。
(450～610行)　クリッピングと描画。
(630～730行)　各軸まわりの回転のサブルーチン。



(写真4-8)



(写真4-9)

(リスト4-5)

```
100 '◆◆ ｽﾞｹｲ ﾉ ｲﾄﾞｳ･ｶｸﾀﾞｲ･ｼｭｸｼｮｳ･ｶｲﾃﾝ / (C) S.Tajima ◆◆
110 '
120 DIM X(100),Y(100),Z(100):CLS:M=1:I=1:ON ERROR GOTO 340
130 READ X:IF X=9999 THEN C=7:GOSUB 350:GOTO 200
140 READ Y,Z:X(I)=X:Y(I)=Y:Z(I)=Z:I=I+1:GOTO 130
150 DATA 700,-90,300,300,-90,300,300,-90,-300,-300,-90,-300,
    -300,-90,300
160 DATA 700,-90,150,300,-90,150,700,-90,0,300,-90,0,700,-90
    ,-150,300,-90,-150
170 DATA 850,-90,300,-150,-90,-300,1000,-90,300,0,-90,-300,1
    150,-90,300
180 DATA 150,-90,-300,9999
190 '+++++++++++++++++++++++++++++++++++++++
200 WHILE K$<>CHR$(13):K$=INKEY$
210 IFK$=CHR$(28)THEN CLS:AX=AX+40:GOSUB 350
220 IFK$=CHR$(29)THEN CLS:AX=AX-40:GOSUB 350
230 IFK$=CHR$(30)THEN CLS:AY=AY+20:GOSUB 350
240 IFK$=CHR$(31)THEN CLS:AY=AY-20:GOSUB 350
250 IFK$="+"THEN CLS:M=M*1.5:GOSUB 350
260 IFK$="-"THEN CLS:M=M*.5:GOSUB 350
270 IFK$=","THEN CLS:BANK=BANK+.3927:GOSUB 350
280 IFK$="="THEN CLS:BANK=BANK-.3927:GOSUB 350
290 IFK$="6"THEN CLS:HEAD=HEAD+.3927:GOSUB 350
300 IFK$="4"THEN CLS:HEAD=HEAD-.3927:GOSUB 350
310 IFK$="8"THEN CLS:PICH=PICH+.3927:GOSUB 350
320 IFK$="2"THEN CLS:PICH=PICH-.3927:GOSUB 350
330 WEND:END
340 RESUME NEXT
350 '+++++++++++++++++++++++++++++++++++++++
360 LINE(0,0)-(639,199),PSET,1,B
370 FOR J=1 TO I-1
380 IF X(J)>=500 THEN 490
390 IF X(J-1)>=500 THEN XX=X(J-1)-1000 ELSE XX=X(J-1)
400 YY=Y(J-1):ZZ=Z(J-1):GOSUB 620:X1=BX:Y1=BY
410 XX=X(J):YY=Y(J):ZZ=Z(J):GOSUB 620:X2=BX:Y2=BY
420 X1=X1*M+AX+320:Y1=100-(Y1*M+AY)*.4495
430 X2=X2*M+AX+320:Y2=100-(Y2*M+AY)*.4495
440 '+++++++++++++++++++++++++++++++++++++++
450 IF X1>0 AND X1<638 AND Y1>0 AND Y1<198 AND X2>0 AND X2<6
    38 AND Y2>0 AND Y2<198 THEN 480
460 IF (X1<0 AND X2<0) OR (X1>638 AND X2>638) OR (Y1<0 AND Y
    2<0) OR (Y1>198 AND Y2>198) THEN 490
```

```
470 GOSUB 510:SWAP X1,X2:SWAP Y1,Y2:GOSUB 510
480 LINE(X1,Y1)-(X2,Y2),PSET,C
490 NEXT:RETURN
500 '+++++++++++++++++++++++++++++++++++++
510 IF X2<1 THEN XX=1:GOTO 540
520 IF X2>638 THEN XX=638:GOTO 540
530 XX=X2
540 IF X2=X1 THEN YY=Y2:GOTO 560
550 YY=Y2-(Y2-Y1)*(X2-XX)/(X2-X1)
560 IF YY<1 THEN YY=1:GOTO 590
570 IF YY>198 THEN YY=198:GOTO 590
580 X2=XX:Y2=YY:GOTO 610
590 IF Y2=Y1 THEN XX=X2:GOTO 580
600 XX=X2-(X2-X1)*(Y2-YY)/(Y2-Y1):GOTO 580
610 RETURN
620 '+++++++++++++++++++++++++++++++++++++++++
630 XX1=XX
640 YY1=YY*COS(PICH)-ZZ*SIN(PICH)
650 ZZ1=YY*SIN(PICH)+ZZ*COS(PICH)
660 XX2=XX1*COS(HEAD)-ZZ1*SIN(HEAD)
670 YY2=YY1
680 ZZ2=XX1*SIN(HEAD)+ZZ1*COS(HEAD)
690 XX3=XX2*COS(BANK)-YY2*SIN(BANK)
700 YY3=XX2*SIN(BANK)+YY2*COS(BANK)
710 ZZ3=ZZ2
720 BX=XX3/(ZZ3*.001+1):BY=YY3/(ZZ3*.001+1)
730 RETURN
```

サンプルデータ

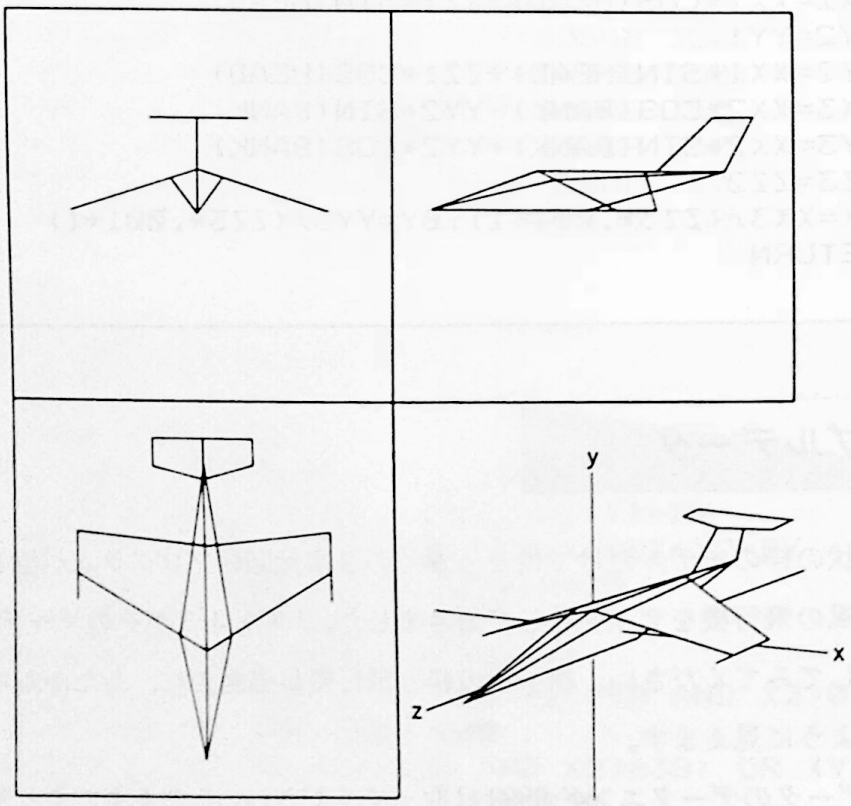
三角や格子状の枠のモデルだけではせっかくの3次元回転プログラムが泣きますので，ビジネスジェット風の飛行機をデータ化しておきました。リスト4-6がそのデータで，リスト4-5につないで実行してみてください。格子状の枠と飛行機が描画され，あたかも地平線上を滑空するジェットのように見えます。

なお，格子データのデータエンド9999は取ってください。そのままですと飛行機のデータは読んでくれません。

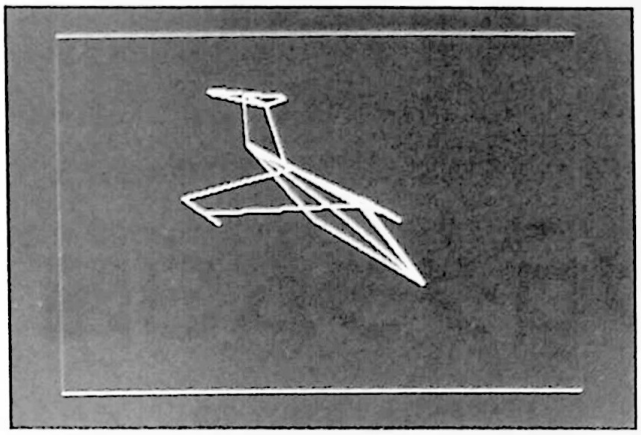
（リスト4-6　飛行機のサンプルデータ）

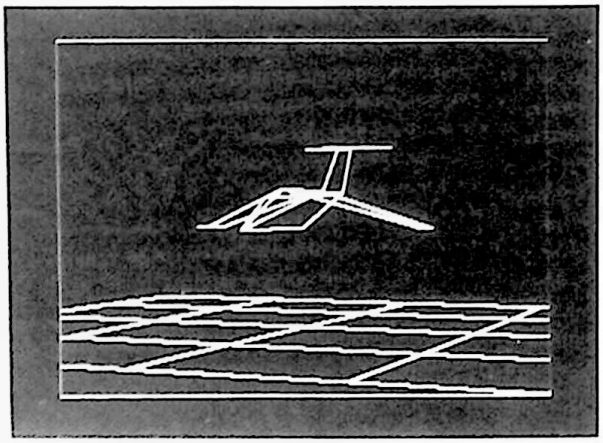
```
1000 DATA 1000,-10,-120,-20,14,-28,0,20,100,20,14,-28,0,-10,-
     120,0,20,-40
1010 DATA 0,20,100,0,-10,0,0,-10,-120:'ﾄﾞｳﾀｲ
1020 DATA 900,-10,0,-100,-10,50,0,20,40,100,-10,50,100,-10,0,
     100,-10,20
1030 DATA 0,20,-40,-100,-10,20:'ｼｭﾖｸ
1040 DATA 1000,20,50,0,60,90,0,60,120,0,20,100:'ﾋﾞﾖｸ
1050 DATA 1000,60,90,-40,60,110,-40,60,120,40,60,120,40,60,11
     0,0,60,90
1060 DATA 9999
```

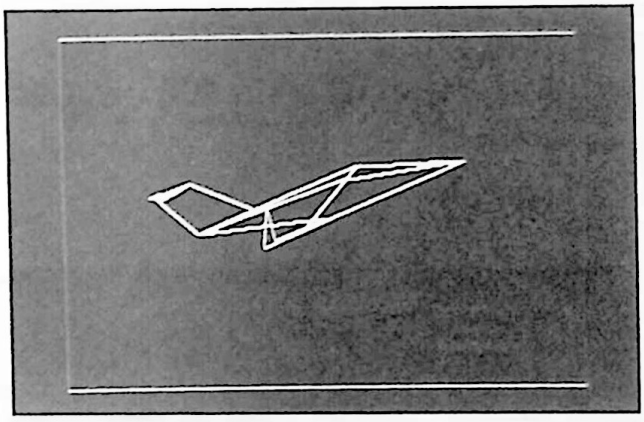
このサンプル以外にも，自分の好きなモデルをデータ化して，3次元グラフィックスの世界を楽しんでください。データ量の多い複雑なモデルを動かすときは，プログラム先頭の配列の数を多めにとることもお忘れなく。

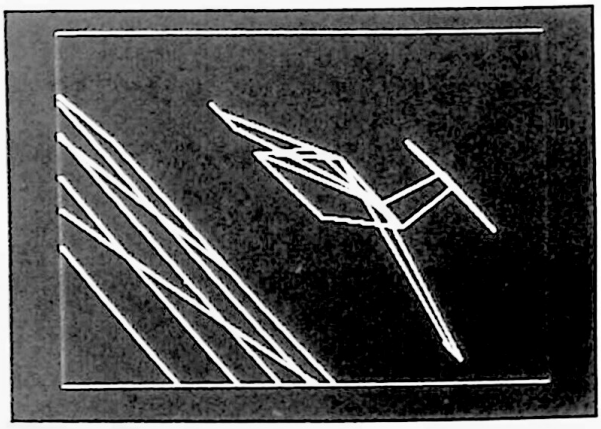


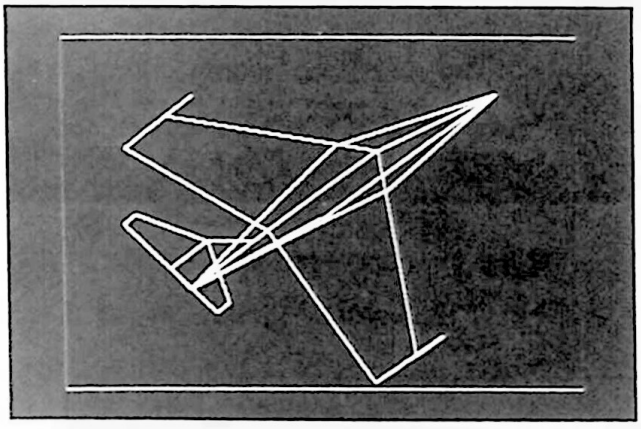
（図4-22　サンプルモデル/飛行機の3面図）

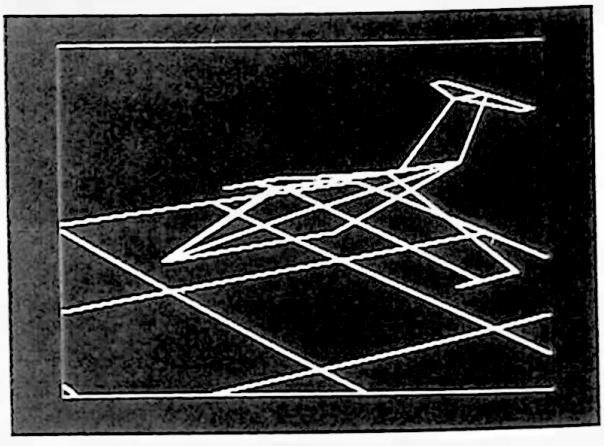




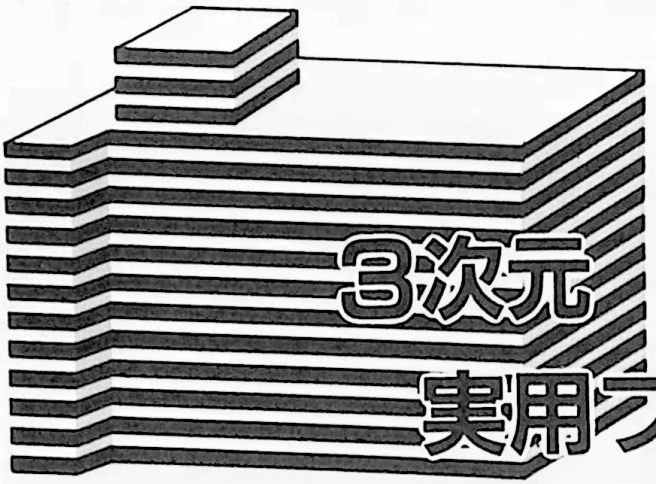








(写真 4-10～15)

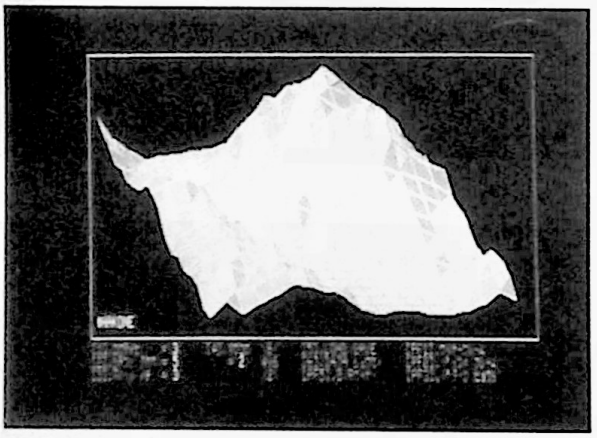


5章

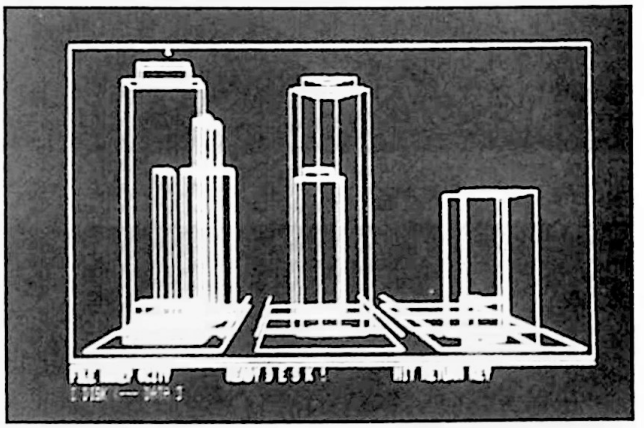
3次元実用プログラム3題

3次元グラフィックスの実用プログラムを紹介します。ひとつは，任意の方向から地形を鳥瞰できる「3次元地形図」，もうふたつは，第4章の理論を元にした「3次元形状作成プログラム」と，そのハイスピードバージョンです。

前記2点のプログラムは，雑誌で「3D GROUND」「3D WORLD」として発表し好評を博したものですが，今回大幅に改訂のうえ本書に発表しました。



(写真5-1　3次元地形図プログラム
3D GROUND)



(写真5-2　3次元形状作成プログラム
3D WORLD)

3次元地形図プログラム（3D GROUND Ver.2）

等高線が記入された地図（地形図）をひと目見ただけで地形の特徴をとらえ，その山容を頭

の中に構築することは，なかなか難しいものです。

空を飛ぶ鳥の目を借りて地上を見おろすかのごとく，地形図を立体化して眺めることが可能ならば，どんなにか素晴しいことでしょう。

「3次元地形図」は，地形図からプロットした標高データを立体的に図化するプログラムで，さまざまな方向から地形を鳥瞰できます。

出力する図の種類は以下の通りです。

(1) ワイヤーフレームによる立体図

(2) ワイヤーフレームに陰影を付けた立体図

(3) 標高別パターンの平面図

(4) 立体視ができるステレオ 立体図

■ プログラムの実行

プログラム「3 DGND 2」をロードします。

RUN ↵とキーインすると，タイトルと共に初期メニューが表示されます。メニューは，

〔1〕 シンキサクセイ

〔2〕 データ LOAD

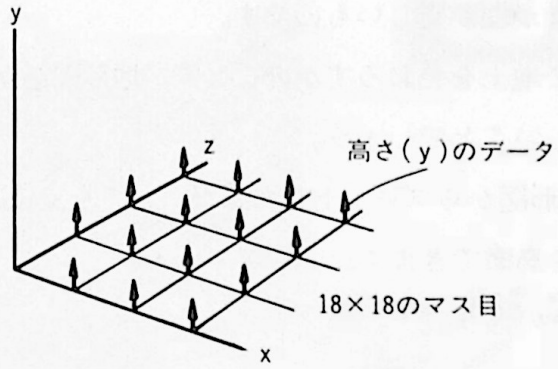
〔3〕 TEST

となっており，1〜3の数字を入力します。〔1〕はキーから新しく地形の標高データを入力することを意味し，〔2〕はすでにデータやディスクなどにSAVEされた標高データを読み込むことを指します。〔3〕はテストデータの作成です。

■ データの新規作成

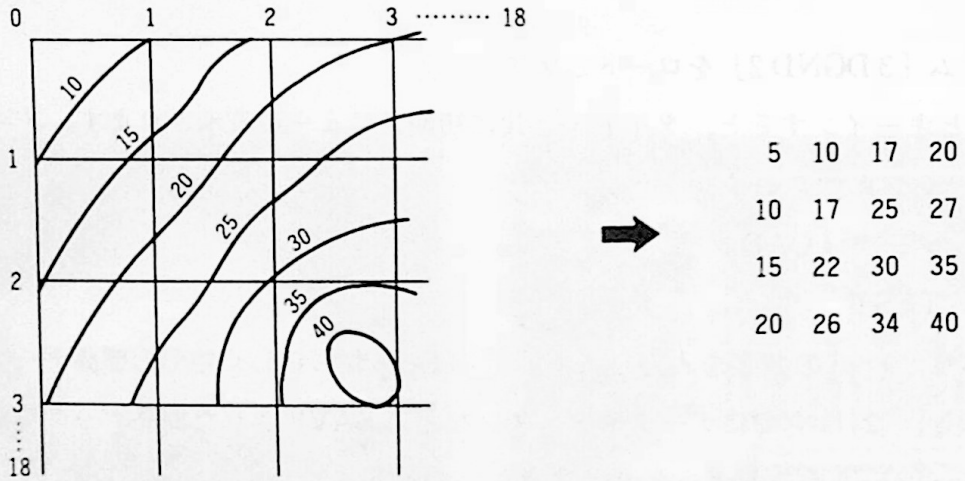
初期メニューで〔1〕を選ぶ（1を押す。↵は不要）と，0〜18の縦横のスケールが表示され，データの入力待ちになります。（写真5-3）

普通3次元グラフィックスは，ひとつの点に対し x，y，z の各座標をプロットしなければならないのですが，このプログラムは標高（Y座標）だけの入力ですみます。標高のデータを求めるには，地形図を用意し任意の区域を18×18個のマス目に仕切ることから始めます。このマス目が先の縦横のスケールに対応し，x および z 座標に相当します。（図5-1）



(図5-1)

標高データはマス目の交点（マス目の中ではありません）の所を等高線から順次読み取ります。少し慣れてくれば，読み取りながらデータをキーインしたほうが効率的ですが，最初のうちは，いったん別紙に書き写してからのほうが安心で間違いありません。(図5-2)



(図5-2　標高データの読み取り)

国土地理院発行の5万分の1や2.5万分の1の地形図を利用する場合，あらかじめ100 mおきに赤ペンなどでなぞっておくと，読み取りが楽になります。また該当区域の地形がマス目の間隔に比べて複雑すぎると，立体図としての精度が落ちますので，範囲をせばめたほうが良い結果が得られます。

データの入力は，

(1)　左上隅から右に入力していきますが，その数値は−998から＋998までの整数に限られます。

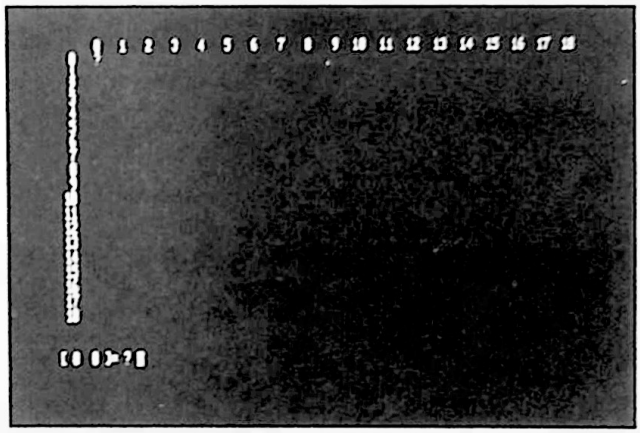
(2) 4桁のデータがあるときは，すべてのデータを10で割ってから入力してください。

(3) 入力の途中で999を入れると訂正ルーチンに入ります。

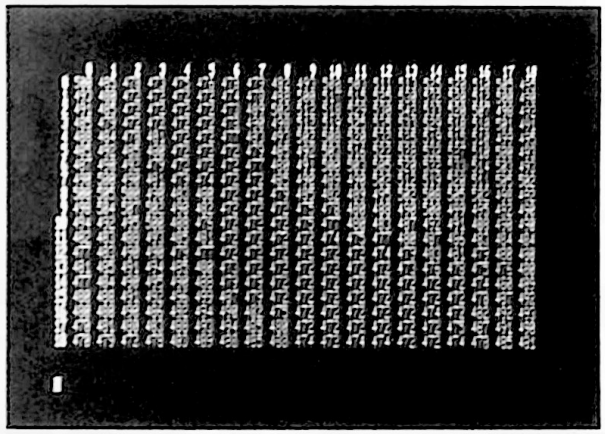
(4) 訂正箇所の横と縦の座標を求めてきますので，場所を指定すると，その位置に？印が表示されます。

(5) 正しいデータを入れ直すと？印に替わって，そのデータが表示され，999を入力した位置から入力の続行となります。

入力すべきデータの数が多いので大変ですが，最後まで頑張ってください。全部のデータの入力をすますと，描画ルーチンに移行します。試験的なデータを入れて，途中で描画させたいときは，−999を入れればそれでも描画ルーチンにジャンプできます。



(写真5-3　入力待ち画面)



(写真5-4　入力済み画面)

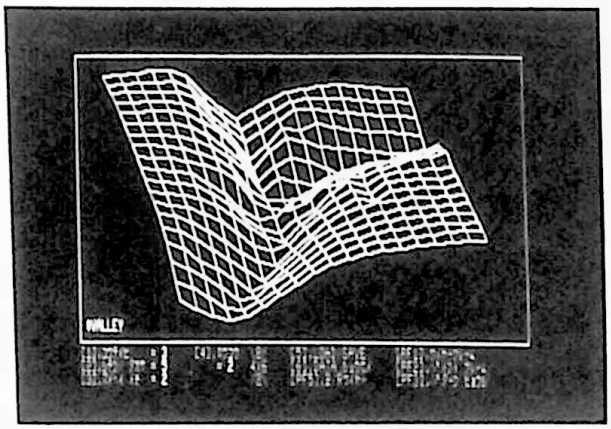
■ 立体図の描画

データを全部入力するか，初期メニューの〔2〕で，データのロードを完了すると，少し間を置いてメインメニューの表示が画面の下方に出ますのでキーで選択して下さい。

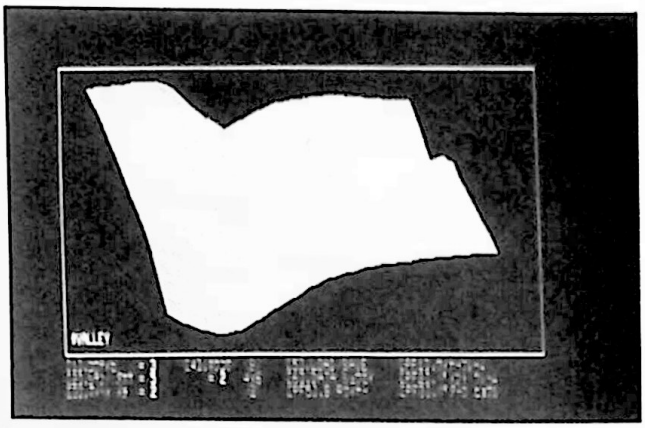
機能選択を操作する主なキーは，テンキーの1〜6，ファンクションキーのPF 1・PF 2・PF 3・PF 5・PF 10です。

PF 1　PF 1はワイヤーフレームで立体図を描きます。

PF 2　PF 2はPF 1で描いた立体図に陰影を付けることができます。ただし，陰影の効果はカラーよりグリーンモニターを使用したほうが自然な表現になります(第1章写真参照)。描画にやや時間がかかるのが難点です。



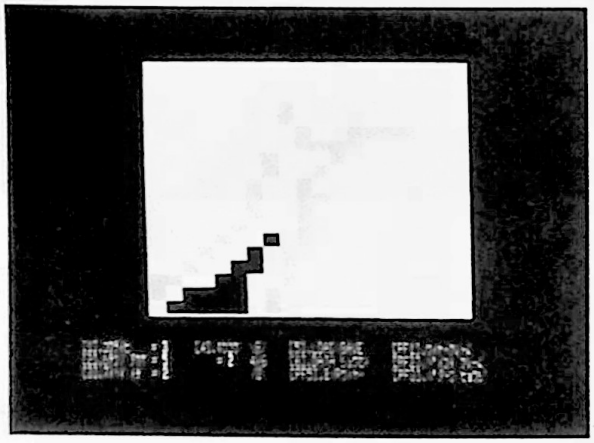
(写真5-5　ワイヤーフレーム)



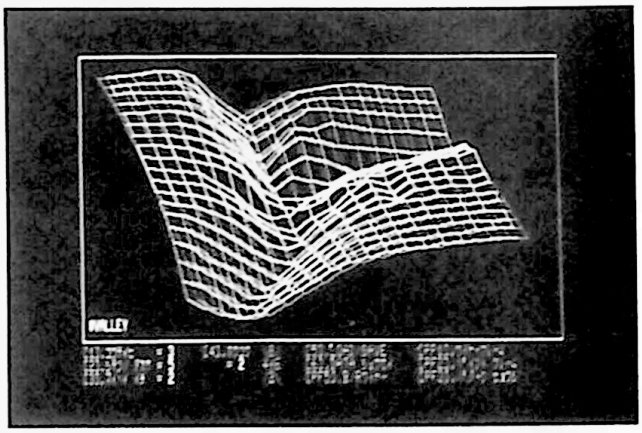
(写真5-6　陰影を付けたフレーム)

PF3　高度別に色分けされたパターンの平面図に描画します。

PF5　立体視を行なうためのステレオ立体図を描画します。青と赤のセロファンで作ためがねで見ると超3次元感覚の立体感が楽しめます。



(写真5-7　高度別パターン)



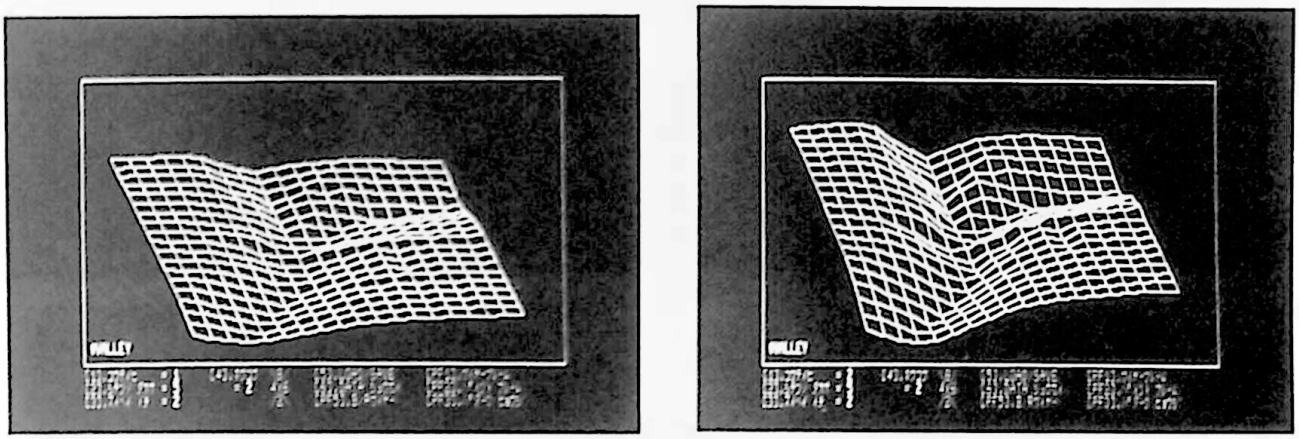
(写真5-8　ステレオ 立体図の描画)

PF10　このキーは，前記PF1，2，3，5の描画を中断させ，メインメニューの入力待ちに戻す働きをします。陰影を付けた立体図描画は時間がかかりますので，描き上がるまで待てないときは，このキーを押してください。

1　テンキーの1を押すと，画面上部に，

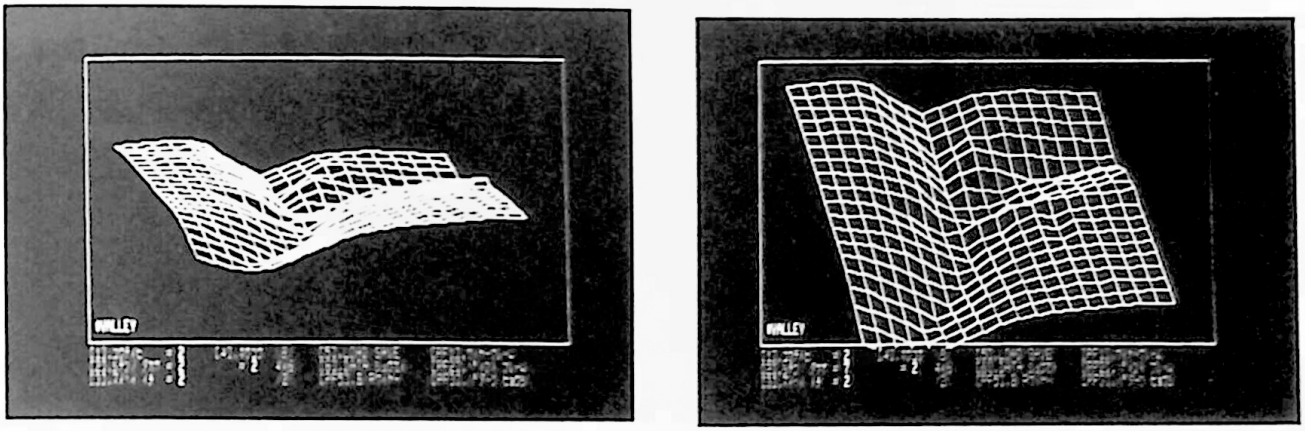
タカサノ　ヒリツ (0　to　8) ?

と，0から8までの数字を求めてきます。入力する数字の大きさに従って，地形の高低比が設定されます。つまり，同じデータでも，なだらかな丘陵表現や，急峻な山岳表現が自在にコントロールできるわけです。普通は，2〜3の入力値をとります。



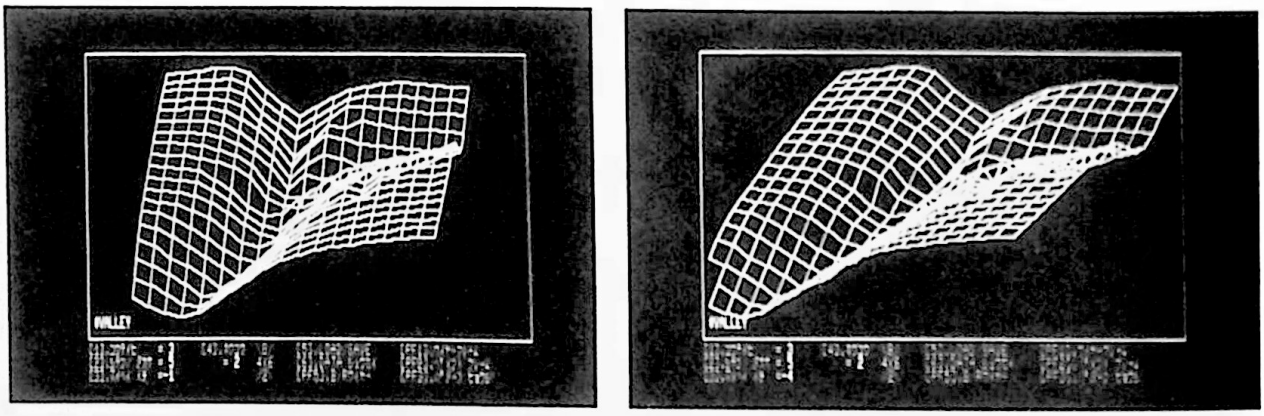
（写真 5-9，5-10　高低比の変化）

2　これは見る位置の高度を設定します。0から8までの値が入力できます。0のときは高度0です。数値が大きくなるにつれて視点が上がり，上から下を見おろす角度が大きくなります。



（写真 5-11，5-12　視点高度の変化）

3　これは，視点の水平位置を決めます。−8から＋8までの値が設定できます。0のときは真正面から見ることになります。



(写真 5-13, 5-14　視点の水平位置の変化)

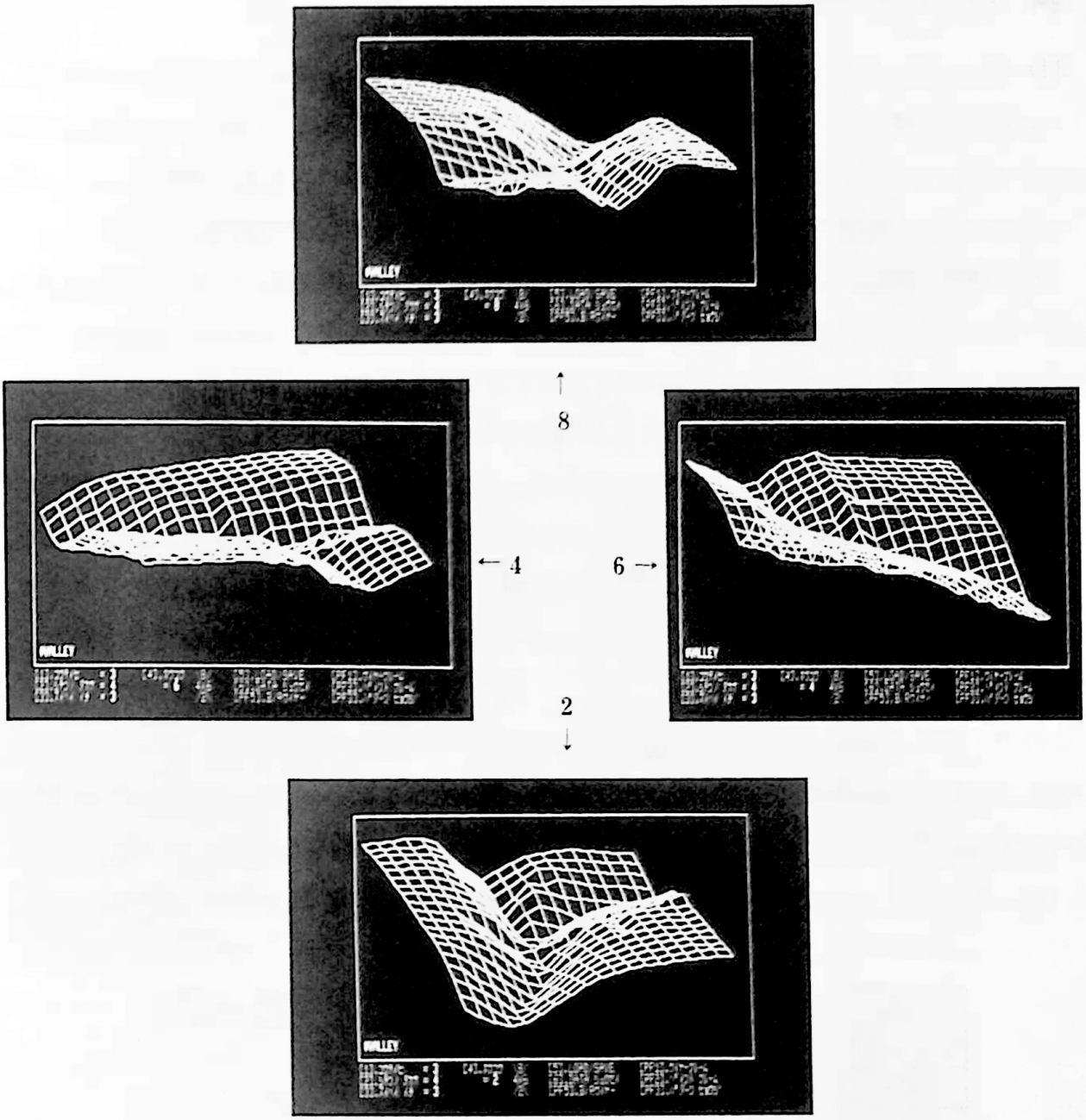
[1], [2], [3] とも，あまり大きな数字を入力すると，隅の方が画面に表示されず，一部不自然な描写になりますので注意が必要です。また，入力値は小数点も受けつけます。

[4]　4を選ぶと，見る方角が変更できます。東西南北の方角指定は，テンキーの配列にならって，2, 6, 8, 4の4つのキーで行ないます。この設定をすると，配列の並べ変えのため，少し間をおいてメニューに戻ります。(写真 5-15～18)

[5]　テープやディスクに対し，データの入出力を実行します。保存か読み込みか聞いてきますので，1か2で答えてください。ファイル名を入力すると転送を開始します。

[6]　データの修正を行ないます。このキーを押すと，全データが表示されますので，誤った箇所の座標を指定し，正しいデータを入れ直してください。入力後，リターンキーを空打ちすると，メインメニューに戻ります。訂正を続行するときは，スペースバーを押してください。

[E]　これはメニューに表示されていませんが，プログラム終了キーです。



(写真 5-15～18　方角の変化)

■ ステレオ立体図

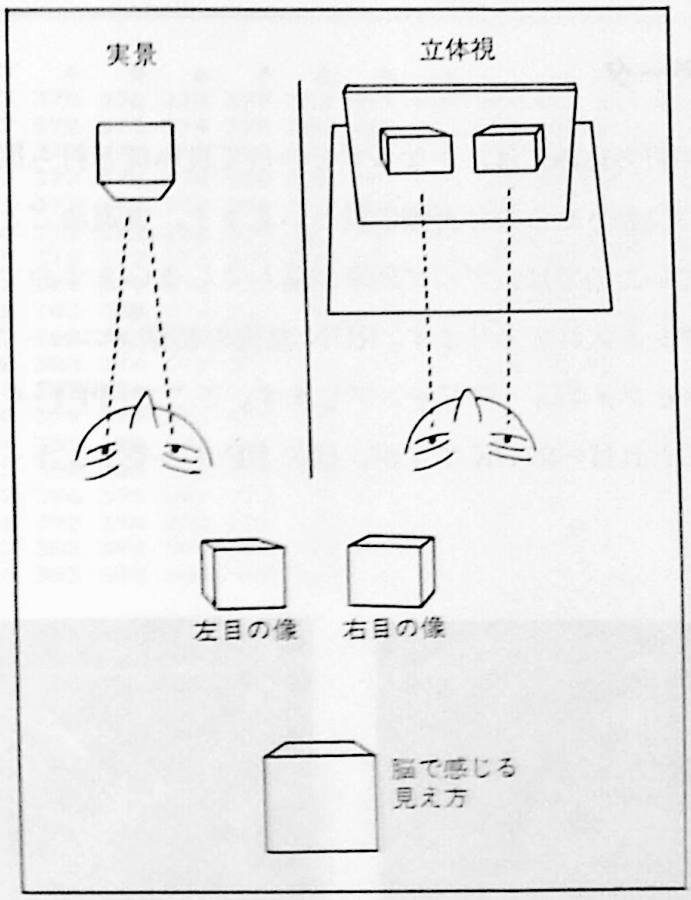
3D ビジョンとか，立体写真などの俗称で呼ばれていますが，原理は航空写真の判読に用いられる立体視と同じです。航空写真の立体視は，60％ずつ平行移動して撮影した同区域の写真を左右に並べ，左目で左の写真を，右目で右の写真を別々に見分け，比高を判断します。この立体視のためには専用のスコープを用いるか，肉眼の場合はある程度の訓練が必要です。

特別な機器や訓練なしにこの立体感を体験するために考えられたのが，いわゆる立体写真で，普通は青と赤のセロファンで作られためがねをかけて対象を見ます。対象は，向きをずらした2枚の絵を青と赤のインキで同一箇所に刷り重ねてありますので，このめがねを通すことにより左目と右目で，別々の絵を見分けることが可能になります。

なぜ立体的に見えるかですが，これは両目を通した2つの像のずれを脳が判断し，遠近感を与えているからです。試しに片目を閉じて，物を取ろうとすると妙につかみづらくなってしまいます。立体視は，角度を少しずらした2枚の平面図形を，強制的に右目，左目に分けて見させています。網膜に映った2つの像を脳は遠近感として判断するため，結果的に立体感が味わえることになります。(図5-3)

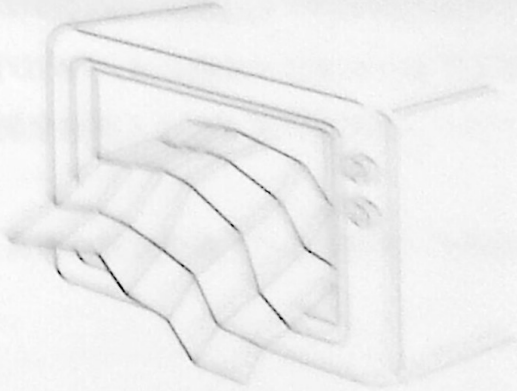
このプログラムでは赤と青のラインを用いて角度のずれた2枚の絵をディスプレイ上に重ね合わせています。場所により，同一ドット上に赤と青が重なる部分が発生しますが，その場合は紫の表示になります。これは，LINE 文の機能である「OR」の論理演算値です(1　OR　2＝3)。赤，青のセロファンを通してこの3色を見ると次表の結果になります。

ラインの色	見え方		
	1(青)	2(赤)	3(紫)
青のライン	見える	見えない	見える
赤のライン	見えない	見える	見える



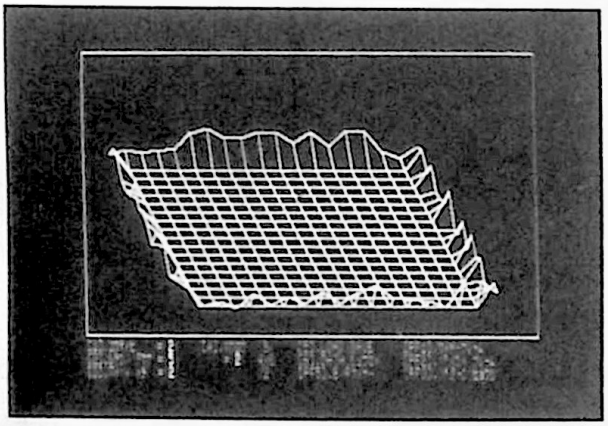
(図 5-3　立体視の原理)

アルで空間的な映像となります。この感覚は文章では適切に表現しにくいため，一応，どのような見え方になるかの図を描いてみました(図 5-4)。まさに，「飛びだす」という表現通りです。ぜひ，この感覚を味わっていただきたいと思います。

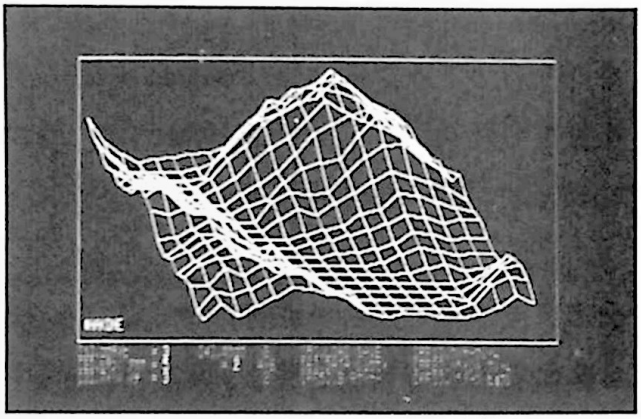


■ テスト用データ

プログラムリストの打ち込みには，かならずといって良いほど打ち間違いのひとつやふたつはあるもので，一発で完動させるのは至難の技といえます。虫退治ごとの動作確認に，データを始めから入れ直していたのでは，デバグ効率が落ちてしまいますので，本プログラムにはテストデータ作成ルーチンを入れてあります。RUN 直後の初期メニューで，3 を選ぶと自動的にデータが生成され，メインメニューにジャンプします。ここで PF 1 を押した場合，写真 5-20 と同じ出力結果が得られれば一応 OK ですが，他の PF キーやテンキーの動作も確認してみてください。



(写真 5-19　テストデータの描画)



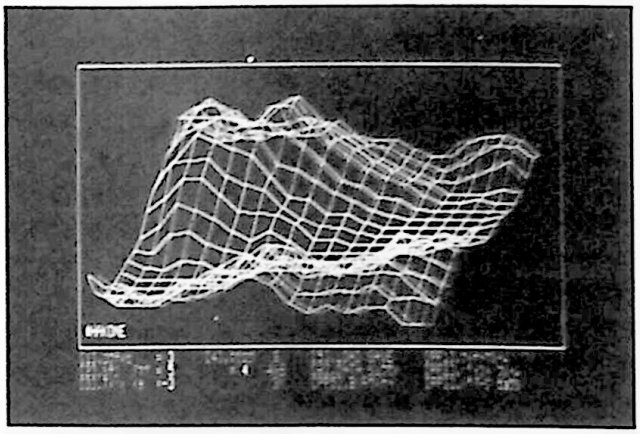
(写真 5-20)

完動が確認されるまでは，1140 行の ON　ERROR　GOTO 文を取っておいたほうが無難です。また，エラー停止後の LIST 表示を実行する前には，「WIDTH　80」，または，「W.　80」を打ち込んでからにしてください。CONSOLE 文による画面制御がとれ，通常のリスト表示になります。

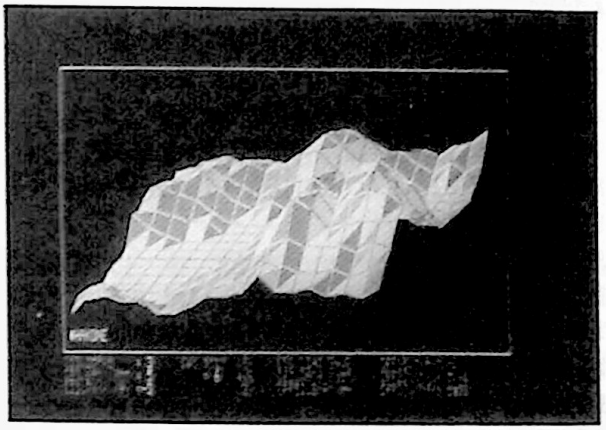
なお，図 5-5 には実際の地形データ例として，箱根近辺の標高データを載せておきました。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	400	385	378	373	375	376	378	379	382	393	400	403	411	413	425	420	410	405	395
1	393	380	372	372	372	372	374	378	388	400	417	415	425	430	432	420	412	400	390
2	391	385	372	372	372	372	373	389	402	412	418	429	430	442	433	420	407	398	392
3	385	379	372	372	372	372	378	390	398	412	419	425	435	433	429	418	410	401	391
4	388	384	378	381	372	372	377	383	393	402	408	420	432	425	419	415	410	399	395
5	385	392	385	380	372	372	372	379	385	392	403	416	422	428	432	427	410	400	393
6	387	390	392	379	372	372	372	377	380	391	400	403	405	420	432	433	414	403	393
7	390	400	390	381	379	374	372	372	385	390	389	390	402	412	419	420	411	405	402
8	400	405	400	390	383	380	372	372	379	385	382	389	395	402	408	405	402	398	392
9	398	404	407	402	390	382	372	372	372	375	378	380	388	390	400	399	395	392	390
10	394	392	395	400	388	376	372	372	372	375	377	380	383	387	392	391	388	390	390
11	382	383	388	393	385	381	372	372	372	372	374	375	377	380	387	386	383	384	384
12	377	383	390	390	379	377	372	372	372	372	372	372	374	379	383	381	379	380	380
13	381	385	391	394	385	380	373	372	372	372	372	372	372	373	376	378	375	376	380
14	377	384	390	395	390	388	379	379	372	372	372	372	372	372	372	372	374	380	389
15	380	388	391	392	396	399	387	379	372	372	372	372	372	372	372	372	378	387	392
16	380	380	390	386	392	394	390	382	375	372	372	372	372	372	374	372	382	392	393
17	370	381	380	383	388	393	394	385	380	372	372	372	372	372	372	375	387	391	390
18	370	379	372	378	383	389	388	385	388	374	372	374	374	372	372	374	378	385	381

(図 5-5　サンプルデータ（箱根近辺））



(写真 5-21)



(写真 5-22)

（リスト5-1　3D　GROUND　Ver. 2）

page 1　[File Name : 3DGND2]

```
1000 CLS:WIDTH80,25
1010 PRINT" ◢■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■◣ "
1020 PRINT"◢                                    ◣"
1030 PRINT"■         3 - D  G R O U N D          ■"
1040 PRINT"■                                     ■"
1050 PRINT"■         ver.2  by S.Tajima          ■"
1060 PRINT"◥                                    ◤"
1070 PRINT" ◥■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■◤"
1080 DIMB(18,18),A(18,18)
1090 DEF FNL1=X1-(X1-X4)*(Y1-I)/(Y1-Y4)
1100 DEF FNL2=X1-(X1-X2)*(Y1-I)/(Y1-Y2)
1110 DEF FNL3=X2-(X2-X4)*(Y2-I)/(Y2-Y4)
1120 DEF FNL4=X2-(X2-X3)*(Y2-I)/(Y2-Y3)
1130 DEF FNL5=X4-(X4-X3)*(Y4-I)/(Y4-Y3)
1140 DEFINT X,Y:ON ERROR GOTO 3000
1150 ONKEY(1) GOSUB 1820
1160 ONKEY(2) GOSUB 1620
1170 ONKEY(3) GOSUB 2520
1180 ONKEY(5) GOSUB 1940
1190 ONKEY(10) GOSUB 2960
1200 FORI=1TO3:KEY(I)OFF:NEXT I
1210 PRINT
1220 PRINT"--------------------------------------"
1230 PRINT"             [1] ･･･ ｼﾝｷ ｻｸｾｲ"
1240 PRINT"             [2] ･･･ DATALOAD"
1250 PRINT"             [3] ･･･ T E S T"
1260 PRINT"--------------------------------------"
1270 PRINT:PRINT" ｱﾀﾗｼｸ ﾃﾞｰﾀ ｦ ｷｰｶﾗ ﾆｭｳﾘｮｸｽﾙﾄｷ ﾊ  1":PRINT
1280 PRINT"  ﾛｰﾄﾞｽﾙﾄｷ ﾊ  2  ｦ ｵｼﾃ ｸﾀﾞｻｲ｡":PRINT:PRINT:GOSUB 2
     980
1290 S$=INKEY$:IF S$="1"THEN GOSUB 2980:GOSUB 2100:GOTO1330
1300 IF S$="2"THEN GOSUB 2980:GOSUB 2840:PRINT:PRINT"ﾁｲｷﾊ ";T
     $;" ﾃﾞｽ｡":PRINT:GOSUB2210:GOTO1330
1310 IF S$="3"THEN GOSUB 2980:GOTO 3020
1320 GOTO 1290
1330 '********** ｼｮｷﾁ **********
1340 HI=2:DA=5:ZI=2:JZ=(ZI+.5)*3:Z=6:F=2:H=HI/HC
1350 '********** MENU **********
1360 WIDTH 80,25:CONSOLE 0,22,0,0
```

```
page 2    [ File Name : 3DGND2 ]
1370 LOCATE 0,22:COLOR 4
1380 GOSUB 2980
1390 PRINT"[1].ｺｳﾃｲﾋ      =          [4].ﾎｳｺｳ  ";CHR$(&HEF);"8";CH
     R$(&HEE);"     [5].LOAD/SAVE      [PF1].ﾜｲﾔｰﾌﾚｰﾑ"
1400 PRINT"[2].ｼﾃﾝﾉ ﾀｶｻ =                =      4";CHR$(&HF0);"6
      [6].DATA ｼｭｳｾｲ     [PF2].ﾍﾟｲﾝﾄ ﾌﾚｰﾑ"
1410 PRINT"[3].ｽｲﾍｲ ｲﾁ  =                          ";CHR$(&HEE);"2";CH
     R$(&HEF);"     [PF5].B/Rﾜｲﾔｰ      [PF3].ﾊﾟﾀｰﾝ ﾋｮｳｼﾞ";
1420 COLOR 7:LOCATE 14,22:PRINT HI;:LOCATE 14,23:PRINT DA;:LO
     CATE 14,24:PRINT ZI;:LOCATE 26,23:PRINT F;
1430 S$=INKEY$:IF S$="e" OR S$="E" THEN FOR I=1 TO 5:KEY(I)OF
     F:NEXT I:KEY(10)OFF:CLS:WIDTH 80:END
1440 FOR I=1 TO 5:KEY(I)ON:NEXT I
1450 IF S$="1" OR S$="2" OR S$="3" OR S$="4" OR S$="5" OR S$=
     "6" THEN 1460 ELSE 1430
1460 GOSUB 2980:LINE(0,0)-(25,1)," ",,B:ON VAL(S$) GOSUB 1480
     ,1510,1530,1550,2770,2280
1470 GOTO 1370
1480 LOCATE 14,22:PRINT" ?    ";:LOCATE 0,0:PRINT"ﾀｶｻﾉ ﾋﾘﾂ (0
     to 8)":INPUT HI :IFHI<0 OR HI>8 THEN 1480
1490 H=HI/HC:LINE(0,0)-(25,1)," ",,B
1500 RETURN
1510 LOCATE 14,23:PRINT" ?    ";:LOCATE 0,0:PRINT"ｼﾃﾝ ﾉ ﾀｶｻ (0
      TO 8)= ":INPUT DA :IF DA<0 OR DA>8 THEN 1510
1520 LINE(0,0)-(25,1)," ",,B:RETURN
1530 LOCATE 14,24:PRINT" ?   " ;:LOCATE 0,0:PRINT"ｽｲﾍｲ ｲﾁ (-8
     TO 8)= ":INPUT ZI :IF ZI<-8 OR ZI>8 THEN 1530
1540 Z=ZI*3:JZ=(ZI+.5)*3:LINE(0,0)-(25,1)," ",,B:RETURN
1550 LOCATE 26,23:PRINT" ?  ":LOCATE 0,0:PRINT"ﾐﾙ ﾎｳｶﾞｸ (2.4.
     8.6)= ":INPUT F :LOCATE 0,0:LINE(0,1)-(25,1)," ",,B
1560 IF F=2 THEN FOR I=0 TO 18:BEEP 1:BEEP 0:FOR J=0 TO 18:A(
     I,J)=B(I,J)-HL:NEXT J,I:GOTO 1610
1570 IF F=6 THEN FOR I=0 TO 18:BEEP 1:BEEP 0:FOR J=0 TO 18:A(
     I,J)=B(18-J,I)-HL:NEXT J,I:GOTO 1610
1580 IF F=8 THEN FOR I=0 TO 18:BEEP 1:BEEP 0:FOR J=0 TO 18:A(
     I,J)=B(18-I,18-J)-HL:NEXT J,I:GOTO 1610
1590 IF F=4 THEN FOR I=0 TO 18:BEEP 1:BEEP 0:FOR J=0 TO 18:A(
     I,J)=B(J,18-I)-HL:NEXT J,I:GOTO 1610
1600 GOTO 1550
```

```
page 3    [ File Name : 3DGND2 ]
1610 RETURN
1620 '********** PAINT MODE **********
1630 KEY(10)ON
1640 FOR I=1 TO 5:KEY(I)OFF:NEXTI
1650 CZ=INT(70-Z*9):W=24:GOSUB 2980
1660 CLS:ZU=0:D=DA:LINE(0,0)-(639,173),PSET,,B
1670 LOCATE 1,20:PRINT T$
1680 FOR X=1 TO 18:FOR Y=1 TO 18
1690 X1=Y*W+ZU+CZ:Y1=-(H*A(X-1,Y-1)-D-110+DA*9):X2=X1+W:Y2=-(
     H*A(X-1,Y)-D-110+DA*9)
1700 X3=X2+Z:Y3=-(H*A(X,Y)-D-DA-110+DA*9):X4=X3-W:Y4=-(H*A(X,
     Y-1)-D-DA-110+DA*9)
1710 IFX1<0OR Y1<0 ORY2<0 ORY2<0 ORY3<0 ORY4<0 ORX2>639 ORX4<
     0 ORX3>639 THEN1810
1720 C=Y1-Y2:C=C+20:C=INT(C*.175):IF C<1 THEN C=1
1730 IF C>6 THEN C=6
1740 ACOLOR=C
1750 GOSUB 2590:CONNECT(X4,Y4)-(X1,Y1)-(X2,Y2),ACOLOR+1
1760 CONNECT(X4,Y4)-(X2,Y2),ACOLOR
1770 C=Y4-Y3:C=C+20:C=INT(C*.175):IF C<1 THEN C=1
1780 IF C>6 THEN C=6
1790 BCOLOR=C
1800 GOSUB 2690:CONNECT(X2,Y2)-(X3,Y3)-(X4,Y4),BCOLOR+1
1810 NEXT Y:D=D+DA:ZU=ZU+Z:NEXT X:COLOR 7:RETURN 1370
1820 '********** WIRE MODE **********
1830 KEY(10)ON
1840 FOR I=1 TO 5:KEY(I)OFF:NEXT I
1850 CLS:GOSUB 2980
1860 CZ=70-Z*9:W=24:ZU=0:D=DA:LINE(0,0)-(639,173),PSET,,B
1870 LOCATE 1,20:PRINT T$
1880 FOR X=1 TO 18:FOR Y=1 TO 18
1890 X1=Y*W+ZU+CZ:Y1=-H*A(X-1,Y-1)+D+110-DA*9:X2=X1+W:Y2=-H*A
     (X-1,Y)+D+110-DA*9
1900 X3=X2+Z:Y3=-H*A(X,Y)+D+DA+110-DA*9:X4=X3-W:Y4=-H*A(X,Y-1
     )+D+DA+110-DA*9
1910 IFX1<0OR Y1<0 ORY2<0 ORY2<0 ORY3<0 ORY4<0 ORX2>639 ORX4<
     0 ORX3>639 THEN1930
1920 CONNECT(X1,Y1)-(X2,Y2)-(X3,Y3)-(X4,Y4)-(X1,Y1)
1930 NEXT Y:D=D+DA:ZU=ZU+Z:NEXT X:RETURN 1370
```

page 4　[File Name : 3DGND2]

```
1940 '********** B/R WIRE MODE **********
1950 KEY(10)ON
1960 FOR I=1 TO 5:KEY(I)OFF:NEXT I
1970 CLS:GOSUB 2980
1980 CZ=70-Z*9:CCZ=70-JZ*9:W=24:ZU=0:ZZU=0:D=DA:LINE(0,0)-(63
     9,173),PSET,,B
1990 LOCATE 1,20:PRINT T$
2000 FOR X=1 TO 18:FOR Y=1 TO 18
2010 X1=Y*W+ZU+CZ:Y1=-H*A(X-1,Y-1)+D+110-DA*9:X2=X1+W:Y2=-H*A
     (X-1,Y)+D+110-DA*9
2020 X3=X2+Z:Y3=-H*A(X,Y)+D+DA+110-DA*9:X4=X3-W:Y4=-H*A(X,Y-1
     )+D+DA+110-DA*9
2030 IFX1<0OR Y1<0 ORY2<0 ORY2<0 ORY3<0 ORY4<0 ORX2>639 ORX4<
     0 ORX3>639 THEN2090
2040 CONNECT(X1,Y1)-(X2,Y2)-(X3,Y3)-(X4,Y4)-(X1,Y1),1,OR
2050 X1=Y*W+ZZU+CCZ:Y1=-H*A(X-1,Y-1)+D+110-DA*9:X2=X1+W:Y2=-H
     *A(X-1,Y)+D+110-DA*9
2060 X3=X2+JZ:Y3=-H*A(X,Y)+D+DA+110-DA*9:X4=X3-W:Y4=-H*A(X,Y-
     1)+D+DA+110-DA*9
2070 IFX1<0OR Y1<0 ORY2<0 ORY2<0 ORY3<0 ORY4<0 ORX2>639 ORX4<
     0 ORX3>639 THEN2090
2080 CONNECT(X1,Y1)-(X2,Y2)-(X3,Y3)-(X4,Y4)-(X1,Y1),2,OR
2090 NEXT Y:D=D+DA:ZU=ZU+Z:ZZU=ZZU+JZ:NEXT X:RETURN 1370
2100 '********** INPUT DATA **********
2110 CONSOLE 22,3,0,0
2120 GOSUB 2310:COLOR 4:LINE(0,22)-(79,23)," ",,BF
2130 FOR I=0 TO 18:FOR J=0 TO 18:LOCATE J*4+6,I+1:COLOR 6:PRI
     NT "?":COLOR 4
2140 LOCATE 0,22:COLOR 7:PRINT"[";I;J;"]= ";:LOCATE 10,22:INP
     UT B(I,J):GOSUB 2980
2150 IF B(I,J)>=1000 OR B(I,J)<=-1000 THENPRINT"ﾃﾞｰﾀﾉ ｹﾀ ｶﾞ ｵ
     ｵｷｽｷﾞﾏｽ。":BEEP:GOTO 2140
2160 IF B(I,J)=999 THENLOCATEJ*4+6,I+1:PRINT"-":GOSUB 2380:GO
     TO2140
2170 IF B(I,J)=-999 THEN B(I,J)=0:GOTO 2210
2180 COLOR 4:LOCATE J*4+3,I+1:PRINTUSING"####";B(I,J)
2190 LINE(0,22)-(20,23)," ",,B
2200 NEXT J,I
2210 '********** ｻｲﾀﾞｲ ｻｲｼｮｳﾁ **********
```

```
page 5    [ File Name : 3DGND2 ]
2220 LOCATE 0,22:PRINT"ｼﾊﾞﾗｸ ｵﾏﾁｸﾀﾞｻｲ。          "
2230 HH=B(0,0):HL=HH:FORI=0TO18:BEEP 1:BEEP 0:FORJ=0TO18:IF H
     H<B(I,J) THEN HH=B(I,J)
2240 IF HL>B(I,J) THEN HL=B(I,J)
2250 NEXT J,I:FOR I=0 TO 18:FOR J=0 TO 18:BEEP 1:BEEP 0
2260 A(I,J)=B(I,J)-HL:NEXTJ,I:HS=HH-HL:HC=INT(HS/20):IF HC=0
     THEN HC=1
2270 RETURN
2280 '********** DATA ｼｭｳｾｲ **********
2290 KEY(10)ON
2300 CONSOLE 22,3,0,0:GOSUB 2310:GOSUB 2350:GOSUB 2380:GOSUB
     2210:CLS:RETURN
2310 '********** DATA SCALE **********
2320 CLS:COLOR 7
2330 FOR I=0 TO 18:LOCATE I*4+3,0:PRINTUSING"####";I;:LOCATE
     0,I+1:PRINTUSING"###";I:NEXT I
2340 RETURN
2350 '********** DATA LIST **********
2360 COLOR 4:FOR I=0 TO 18:FOR J=0TO18:LOCATE J*4+3,I+1:PRINT
     USING"####";B(I,J)
2370 NEXT J,I:RETURN
2380 '********** ｼｭｳｾｲ **********
2390 COLOR 6
2400 GOSUB 2980:LINE(0,22)-(79,24)," ",,BF:LOCATE 0,22:PRINT"
     [ DATA ｼｭｳｾｲ ]"
2410 LOCATE 25,22:PRINT"ﾖｺ ｻﾞﾋｮｳ=";:LOCATE 33,22:INPUT YO:GOS
     UB 2980:IF YO<0 OR YO>18 THEN BEEP:GOTO 2400
2420 LOCATE 25,23:PRINT"ﾀﾃ ｻﾞﾋｮｳ=";:LOCATE 33,23:INPUT TA:GOS
     UB 2980:IF TA<0 OR TA>18 THEN BEEP:GOTO 2420
2430 LOCATE YO*4+4,TA+1:COLOR 2:PRINT"  ?":COLOR 6
2440 LOCATE 45,23:PRINT"data=";:INPUT B(TA,YO):IF B(TA,YO)>=1
     000 THEN BEEP:GOTO 2440
2450 COLOR 2:GOSUB 2510:PRINTUSING"####";B(TA,YO);:COLOR 6
2460 LOCATE45,24:PRINT"data ﾉ ｼｭｳｾｲﾖ ﾂﾂﾞｹﾙ = [SPACE KEY]    ｵ
     ﾜﾙ = [RETURN KEY]"
2470 S$=INKEY$:IF S$=" " THEN COLOR 4:GOSUB 2510:PRINTUSING"#
     ###";B(TA,YO);:GOTO 2390
2480 IF S$=CHR$(13)THEN COLOR 4:GOSUB 2510:PRINTUSING"####";B
     (TA,YO);:GOTO 2500
```

page 6 [File Name : JDGND2]

```
2490 GOTO 2470
2500 LINE(0,22)-(79,24)," ",,BF:RETURN
2510 LOCATE YO*4+3,TA+1:RETURN
2520 '********** ﾊﾟﾀｰﾝ **********
2530 FOR I=1 TO 5:KEY(I)OFF:NEXT:KEY(10)ON
2540 CLS:GOSUB 2980:LINE(103,7)-(560,160),PSET,,B
2550 FOR I= 0TO 18:FOR J=0 TO 18
2560 C=INT(A(I,J)/HC/2.8):IF C>7 THEN C=7
2570 LOCATE J*3+13,I+1:COLOR C:PRINT "■■";
2580 NEXTJ,I:RETURN 1370
2590 '******** LINE PAINT ********
2600 IF Y4<=Y1-((Y2-Y1)*(X1-X4))/(X2-X1) THEN 2670
2610 IF Y4=Y1 THEN FORI=Y2 TO Y1:CONNECT(FNL3,I)-(FNL2,I),ACO
     LOR:NEXT:GOTO2670
2620 IF Y1=Y2 THEN FORI=Y1 TO Y4:CONNECT(FNL1,I)-(FNL3,I),ACO
     LOR:NEXT:GOTO2670
2630 IF Y4=Y2 THEN FORI=Y1 TO Y4:CONNECT(FNL1,I)-(FNL2,I),ACO
     LOR:NEXT:GOTO2670
2640 IF Y2>Y4 THEN FORI=Y1 TO Y4:CONNECT(FNL1,I)-(FNL2,I),ACO
     LOR:NEXT:FOR I=Y4 TO Y2:CONNECT(FNL3,I)-(FNL2,I),ACOLOR:N
     EXT:GOTO 2670
2650 IF Y2>Y1 AND Y2<Y4 THEN FOR I=Y1 TO Y2:CONNECT(FNL1,I)-(
     FNL2,I),ACOLOR:NEXT:FOR I=Y2 TO Y4:CONNECT(FNL1,I)-(FNL3,
     I),ACOLOR:NEXT:GOTO 2670
2660 IF Y1>Y2 THEN FORI=Y2 TO Y1:CONNECT(FNL2,I)-(FNL3,I),ACO
     LOR:NEXT:FOR I=Y1 TO Y4:CONNECT(FNL1,I)-(FNL3,I),ACOLOR:N
     EXT:GOTO 2670
2670 RETURN
2680 '******** LINE PAINT BCOLLOR ********
2690 IF Y2>=Y4-((Y3-Y4)*(X4-X2))/(X3-X4) THENGOTO2760
2700 IF Y4=Y3 THEN FORI=Y2 TO Y4:CONNECT(FNL3,I)-(FNL4,I),BCO
     LOR:NEXT:GOTO2760
2710 IF Y4=Y2 THEN FORI=Y4 TO Y3:CONNECT(FNL5,I)-(FNL4,I),BCO
     LOR:NEXT:GOTO2760
2720 IF Y2=Y3 THEN FORI=Y2 TO Y4:CONNECT(FNL3,I)-(FNL5,I),BCO
     LOR:NEXT:GOTO2760
2730 IF Y3<Y4 THEN FORI=Y2 TO Y3:CONNECT(FNL3,I)-(FNL4,I),BCO
     LOR:NEXT:FOR I=Y3 TO Y4:CONNECT(FNL3,I)-(FNL5,I),BCOLOR:N
     EXT:GOTO 2760
```

```
page 7    [ File Name : 3DGND2 ]
2740 IF Y4>Y2 AND Y4<Y3 THEN FOR I=Y2 TO Y4:CONNECT(FNL3,I)-(
     FNL4,I),BCOLOR:NEXT:FOR I=Y4 TO Y3:CONNECT(FNL5,I)-(FNL4,
     I),BCOLOR:NEXT:GOTO 2760
2750 IF Y4<Y2 THEN FOR I=Y4 TO Y2:CONNECT(FNL5,I)-(FNL3,I),BC
     OLOR:NEXT:FOR I=Y2 TO Y3:CONNECT(FNL5,I)-(FNL4,I),BCOLOR:
     NEXT:GOTO 2760
2760 RETURN
2770 '********** LOAD SAVE **********
2780 KEY(10)ON
2790 CLS
2800 PRINT"      ( DATA ﾉ ﾎｿﾞﾝ ) = [1]       ( ﾃﾞｰﾀ ﾉ ﾖﾐｺﾐ ) =
     [2]"
2810 S$=INKEY$:IF S$="1"THEN GOSUB 2890:RETURN
2820 IF S$="2"THEN GOSUB 2840:PRINT:GOSUB 2210:RETURN 1330
2830 GOTO 2810
2840 '********** DATA LOAD **********
2850 INPUT"ﾖﾐｺﾐ FILE NAME ";T$:OPEN"I",#1,T$
2860 INPUT#1,T$:FOR I=0 TO 18:FOR J=0 TO 18
2870 INPUT#1,B(I,J):B(I,J)=B(I,J)+100
2880 NEXT J,I:  CLOSE#1:RETURN
2890 '********** DATA SAVE **********
2900 PRINT:INPUT"ﾎｿﾞﾝ ｽﾙ FILE NAME ";T$:IF LEN(T$)>8 THEN BEE
     P:GOTO 2900
2910 OPEN"O",#1,T$
2920 PRINT#1,T$:FOR I=0 TO 18:FOR J=0 TO 18
2930 PRINT#1,B(I,J)
2940 NEXT J,I:  CLOSE#1:RETURN
2950 '********************************
2960 RETURN 1370
2970 '********** BEEP **********
2980 BEEP 1:FOR BE=0 TO 30:NEXT BE:BEEP 0:RETURN
2990 '********** ERROR **********
3000 BEEP:PRINT"ｴﾗｰ No";ERR;" ｶﾞ ﾊｯｾｲ ｼﾏｼﾀ。 ﾒﾆｭｰ ﾆ ﾓﾄﾞﾘﾏｽ。"
3010 FOR TI=0 TO 8000:NEXT:RESUME 1350
3020 '********** TEST **********
3030 PRINT"DATA ｻｸｾｲ ﾁｭｳ":PRINT
3040 FOR I=0 TO 18:B(I,0)=RND*10:B(I,18)=RND*10
3050 FOR J=0 TO 18:B(0,J)=RND*10:B(18,J)=RND*10
3060 NEXT J,I:GOSUB 2210:GOTO 1330
```

3 次元モデル形状作成プログラム（3D WORLD Ver.2）

このプログラムは，第4章の3次元プログラムを実用向きに発展させたもので，3面図のデータをもとにさまざまな角度からの立体図を作成します。

建築のパースや，機械のテクニカルイラスト等は，人間の手によって描き起こすのが普通ですが，これは多くの時間と大変な熟練を必要とします。

本プログラムは，3面図があれば僅かの根気で正確な立体図が描画できます。また，座標データの入力も，省力化モードを設定し，能率的な作業ができるように工夫してみました。

実用として，また趣味としても充分楽しめますので，気軽に3D.C.Gの扉をたたいてください。

■ プログラムの構造と機能

本プログラムは，大きく分けると以下の3つのモードによりなりたっています。

(1) メインメニュー（メニューモード）

(2) 3面図表示とデータ入力（3面図モード）

(3) 立体図描画（アクションモード）

各モードの機能は図5-6の通りです。図中で，RETは↵リターンキー，BSは⇦バックスペースキーを表わします。入力は大文字しか受けませんので，CAPSロックをかけておいてください。

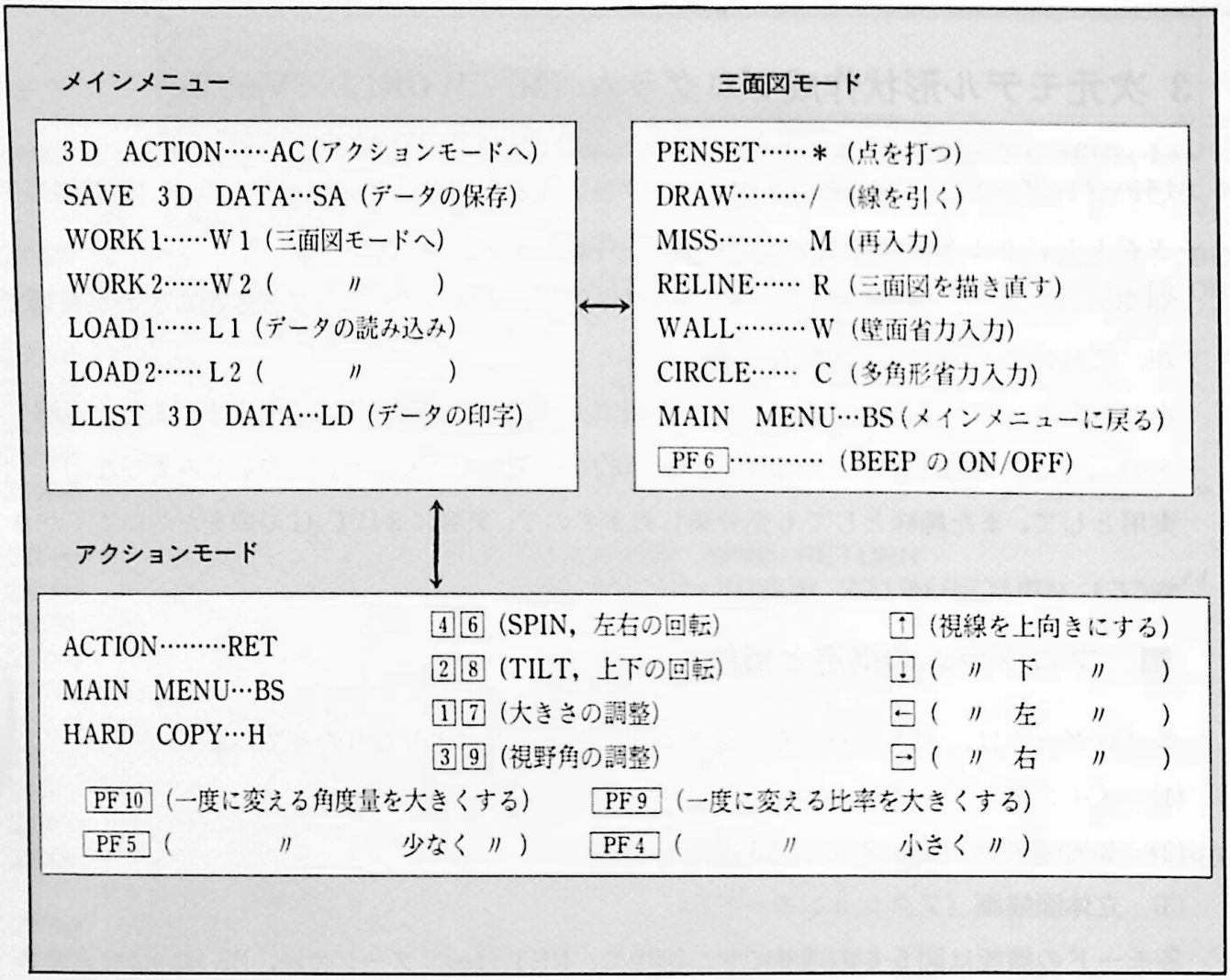
● メインメニュー

「3DWLD2」をロードします。

RUN直後の画面表示はメインメニューから始まります。ここでは，各モードへの移行や，データの入出力，印字を選択します。

●3Dアクション

立体図を描くアクションモードを選択します。メインメニューでAC↵とキーインするとこ



(図5-6　プログラムの構造と機能)

のモードに移行しますが，詳しくは後で述べます。

● SAVE 3D DATA

図形のデータをディスクに記録します。ファイル名を入力し，ディスクをONにしてからリターンキーをたたくと転送を開始します。

● WORK 1

3面図モードへ移行します。新しくデータを作成するときはこれを選択します。

● WORK 2

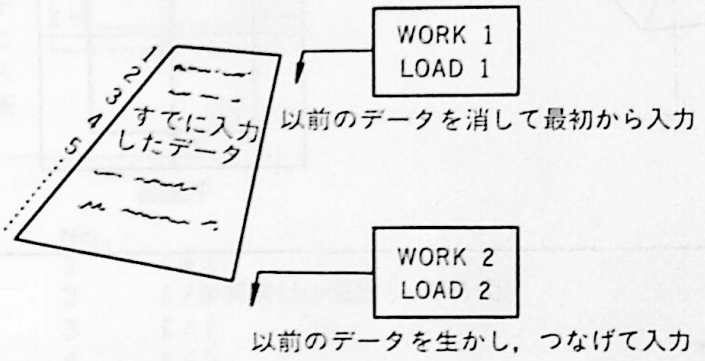
3面図モードへ移行します。WORK 1は新規にデータの作成を行なうのに対し，WORK 2は前のデータを消さずにデータの追加ができます。

● **LOAD 1**

データをディスクから読み込みます。保存したときのファイル名を入力し，リターンキーをたたくと転送を開始します。転送中はデータのステップ数もリアルタイムで表示されます。(前のデータは消えます)

● **LOAD 2**

データをディスクから読み込みますが，WORK 2 と同様に，前のデータに連結して読み込みます。(図 5-7)



(図 5-7　入力 1 と 2 の違い)

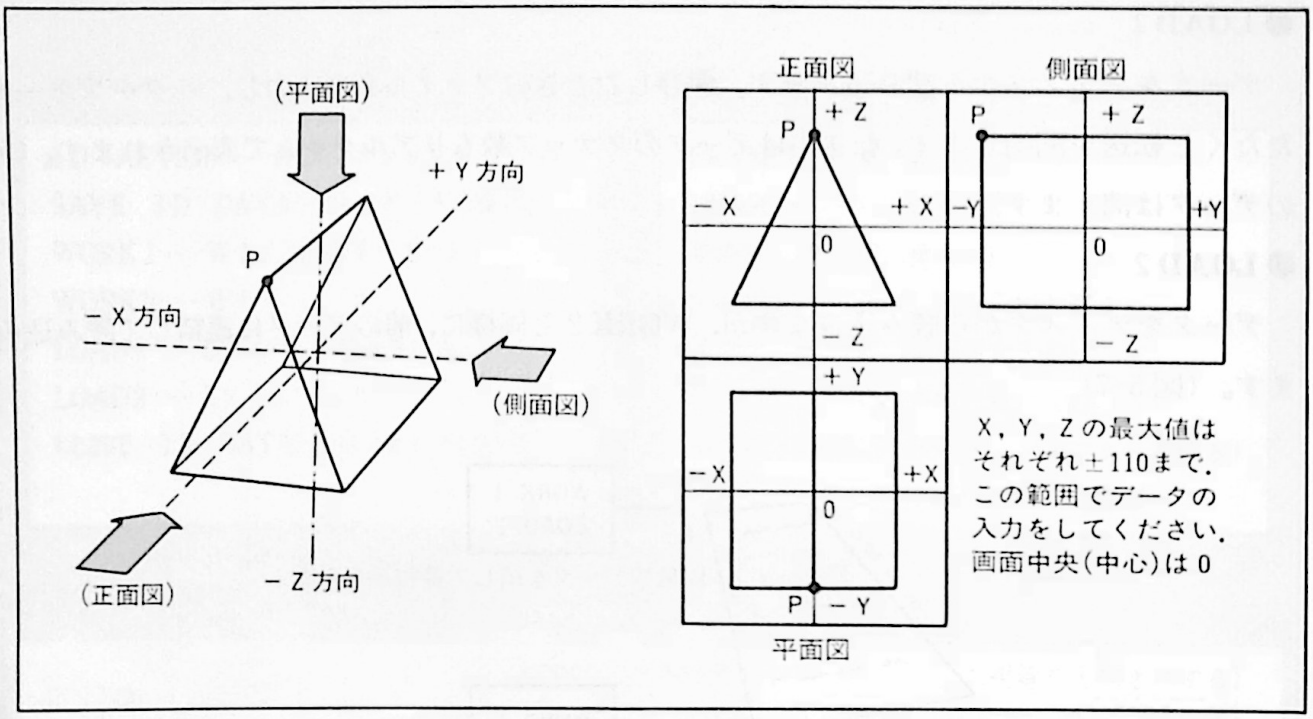
● **LLIST　3 D　DATA**

プリンタに X，Y，Z 座標の順でデータを印字します。

● **3 面図モード**

メインメニューで WORK 1 か WORK 2 を選ぶとこのモードに入り，3 面図を描画し，入力を促します (図 5-9)。座標と 3 面図の位置関係は図 5-8 のようになっていますが，第 4 章で述べた Y と Z の位置とは反対になっていますので注意してください。このプログラムでは，X を幅，Y を奥行，Z を高さとしてとらえています。

データの入力は，簡単な立方体のようなものでしたらディスプレイを見ながらでも可能ですが，あらかじめ 3 面図を用意し，ディスプレイの 3 面図と比較・確認しながら作業を進めたほうが間違いも少なく，能率的です。



(図5-8　3面図の位置関係)

● **PENSET** …… ＊

BEEPの軽い断続音とともに≪印が，それぞれの入力モードを移動し始めると，「入力してください」という合図です。

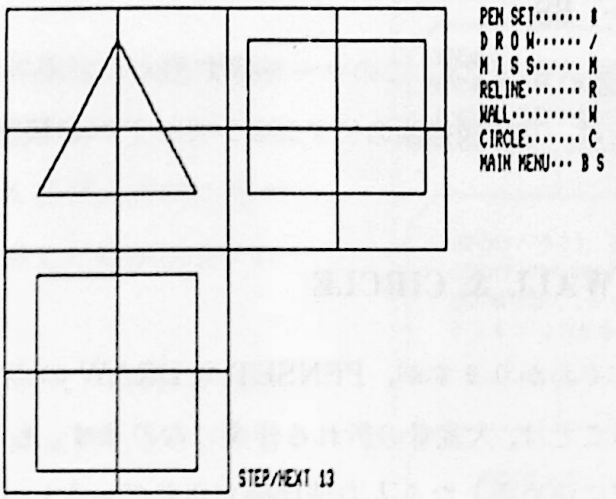
[＊]キーを押すと，≪印はPENSETの位置で停止し，点の座標入力です。X，Y，Zの順で座標値を聞いてきますので，順に入力します。入力ごとに「ピッ」と入力確認の音がでますが，入力値が±110を超えると「ピピー」というBEEPになり再入力となります。

● **DRAW** ……/

[/]は線を引かせるキーです。ひとつ前に入力した座標から，このキーで入力する新座標値まで線を引きます。このキーの入力を繰り返すと，次々と線を結ぶことができます。いったん線を切って，新たな位置から描かせるときは，PENSETの[＊]キーから始めます。

PENSETはペンを上げ，指定された位置でペンを下げ，点を打つことを意味し，DRAWは下がったままのペンを次の座標まで移動させ，線を引くと覚えてください。

PENSETとDRAWを使った例として，図5-9の三角柱のデータを図5-10に入れてあります。入力の確認または練習として利用してください。



(図 5-9 三面図モード)

No.		(X)	(Y)	(Z)
1	[*]	0.0	-90.0	80.0
2	[/]	-80.0	-90.0	-60.0
3	[/]	80.0	-90.0	-60.0
4	[/]	0.0	-90.0	80.0
5	[/]	0.0	90.0	80.0
6	[/]	-80.0	90.0	-60.0
7	[/]	80.0	90.0	-60.0
8	[/]	0.0	90.0	80.0
9	[*]	-80.0	90.0	-60.0
10	[/]	-80.0	-90.0	-60.0
11	[*]	80.0	90.0	-60.0
12	[/]	80.0	-90.0	-60.0

(図5-10 三角柱のデータ)

● MISS …… M

間違えて入力してしまったときは，Mキーを押すとカウントがひとつ戻るとともに，3面図に描画されてしまった点，あるいは線が消えます。このキーを押し続けると，さらにカウントが戻りますので，任意のところからの再入力が可能です。

● RELINE …… R

Rキーは3面図を修復します。前記のMISSを繰り返すと，残って欲しい線や点も消去される場合があります。Rキーは乱れた3面図を初めから描き直してくれます。

● MAIN　MENU …… BS

BS は[⇦]バックスペースキーです。このキーを押すとメインメニューに戻ります。入力したデータの保存や印字または，立体図描画のアクションモードへの移行は[←]キー経由で行ないます。

● 省力的入力法　WALL & CIRCLE

実際にやってみるとすぐわかりますが，PENSET や DRAW の点や線だけで，X，Y，Z の3座標値を入力し続けることは，大変骨の折れる作業になります。もっと大型のシステムでも，この点は課題らしく今だに決め手となる入力法は確立されていないようです。

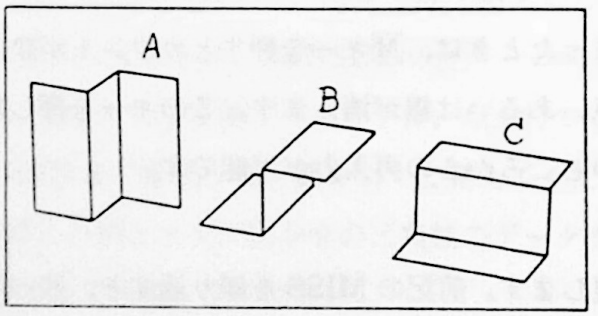
本プログラムでは2つの簡易入力法を取り入れました。あらゆる場合を想定すると，この2つの入力法だけでは対応し切れなくなりますが，そのときは PENSET と DRAW の入力で補ってください。

● WALL …… W

連続した壁面の入力が容易にできます。間取図などの立体図を入力するときは特に威力を発揮します。

[W]キーを押すと，入力したい壁の断面の形が3面図のうちどれに対応（投影）するか聞いてきます。平面図を指定した場合には，壁の高さ(Z1，Z2)はどこからどこまでかを求めてきますので，それぞれの値を入力してください（正面図を指定したときは奥行を，側面図のときは幅を聞いてきます。以後，平面図入力を前提に説明を続けます)。

次に，壁の出発点の X，Y 座標を聞いてきます。この入力をすませると，また X，Y 座標を求めてきますので，前記の DRAW と同様，線をつなげる要領で入力を続けます。ディスプ



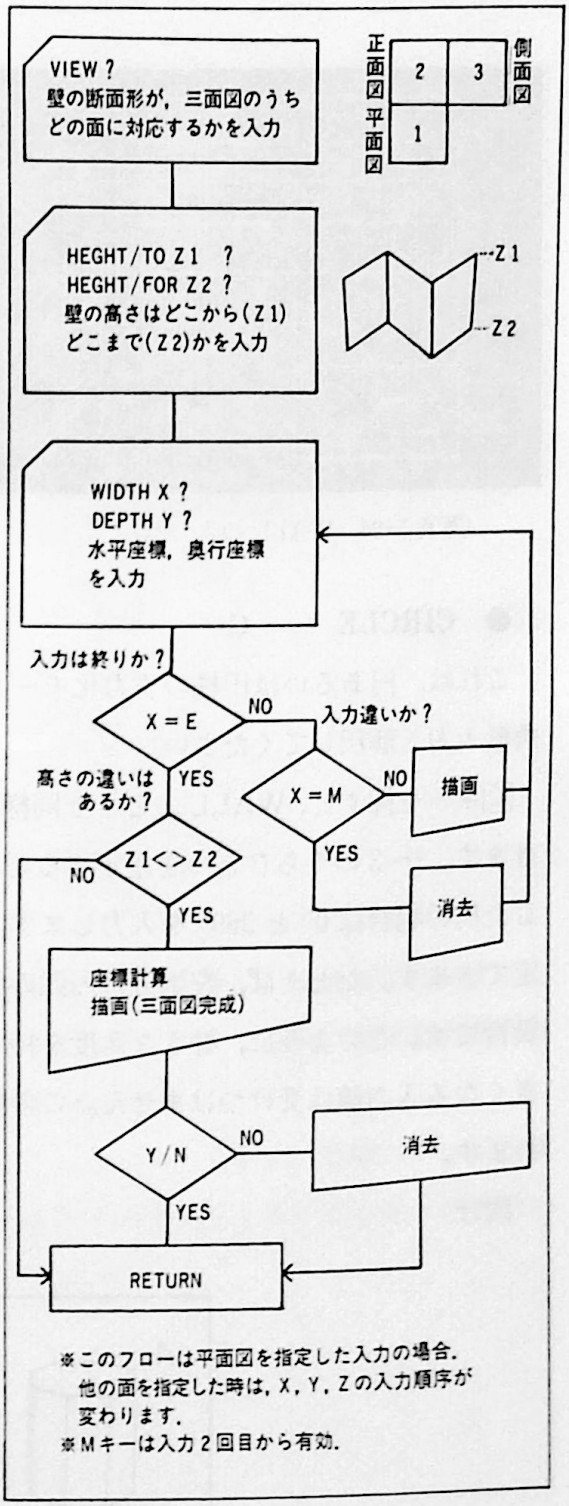
A：平面図入力
B：正面図入力
C：側面図入力

(図5-11　WALL の3面図対応列)

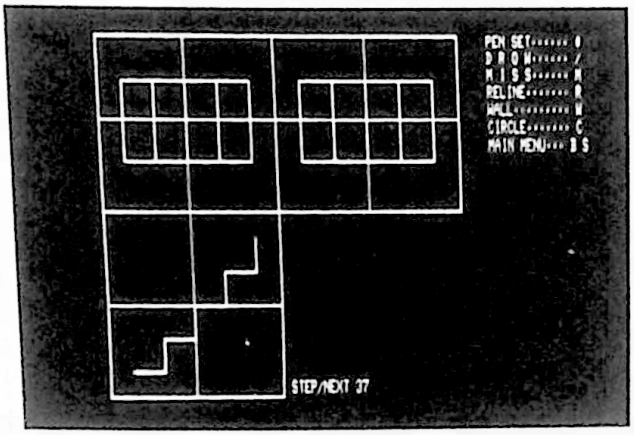
レイの平面図には，入力するたびに壁の断面が描画され，入力値の確認が行なえます。間違えてしまった場合は，X 座標入力時に[M]キーを押してください。間違えた線が消去され再入力になります。

入力すべき壁が終わったら，END を意味する[E]キーを X 座標入力時に押します。すると座標計算が始まり，完成された 3 面図が描画されます。描画が終わると OK かどうかの確認が入りますので，[Y]か[N]で答えてください。[N]を押すと，描画された壁の三面図は消去されます。

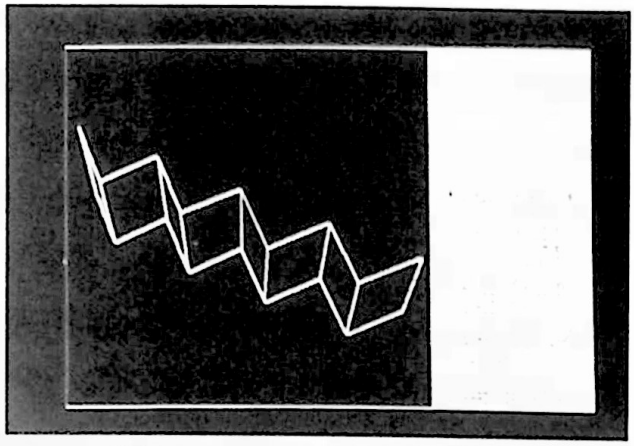
なお，高さ(Z1, Z2)を求めてきたときに同じ数値を入力すると，壁の幅が無くなりますので，高さが一定の連続した線の入力になります。



(図 5-12　WALL モードの入力手順)



(写真 5-23　WALL の入力例)



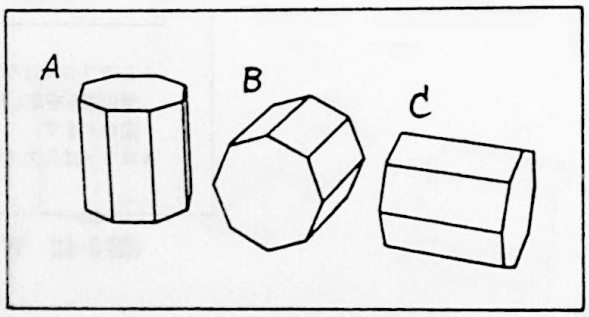
(写真 5-24　WALL の立体図描画例)

● **CIRCLE …… C**

これは，円あるいは円柱の省力化モードです。CIRCLE(円)としてありますが，正確には多角形入力と解釈してください。

Cキーを押すと，WALL のときと同様，円の形が 3 面図のうちどの面に投影するのか聞いてきます。1～3 のうちひとつを指定すると，次は円の開始位置と終了位置を決めます。普通の閉じた円の場合は 0° と 360° を入力します。入力する角度を変えることにより，任意の円弧が設定できます。たとえば，左半分の円弧の場合は 90° と 270°，右半分の円は 270° と 450° という具合です。このように，好きな角度を持つ円弧を指定できますが，開始角度が終了角度より大きくなる入力値は受けつけません。この指定を行なうと，画面下部に指定された円弧が表示されます。

次は



A：平面図入力
B：正面図入力
C：側面図入力

(図 5-13　CIRCLE の 3 面図対応例)

DIVIDED ?

と，円を何度ずつに分けるのかを聞いてきます。これにより，多角形の角数を決めます。前記で0°～360°の閉じた円を指定したときに，ここで45°を入力すると正8角形，15°にすると正24角形になります。小さい角度を指定するほど，角数の多い多角形になりますが，その分大量にメモリを消費します。画面上では，ドットの分解能の関係もあって16角形ぐらいで円に見えてきますので，メモリ効率を考えて適度の角度を指定してください。メモリの消費ステップは多角柱にした場合，(角数＋1)×4となります。この入力時にNキーを押すと，円の開始角度の指定からやり直すことができます。

以上の設定が終わると次は，円柱の高さ，中心座標，半径の入力です。

TO Z1 ? FOR Z2 ?

と円柱の高さを求めてきますので，それぞれ入力します。円柱ではなく，ただの円にしたいときは，Z1，Z2とも同じ数値を入れてください。

次に，

WIDTH/CNT. X ? DEPTH/CNT. Y ?

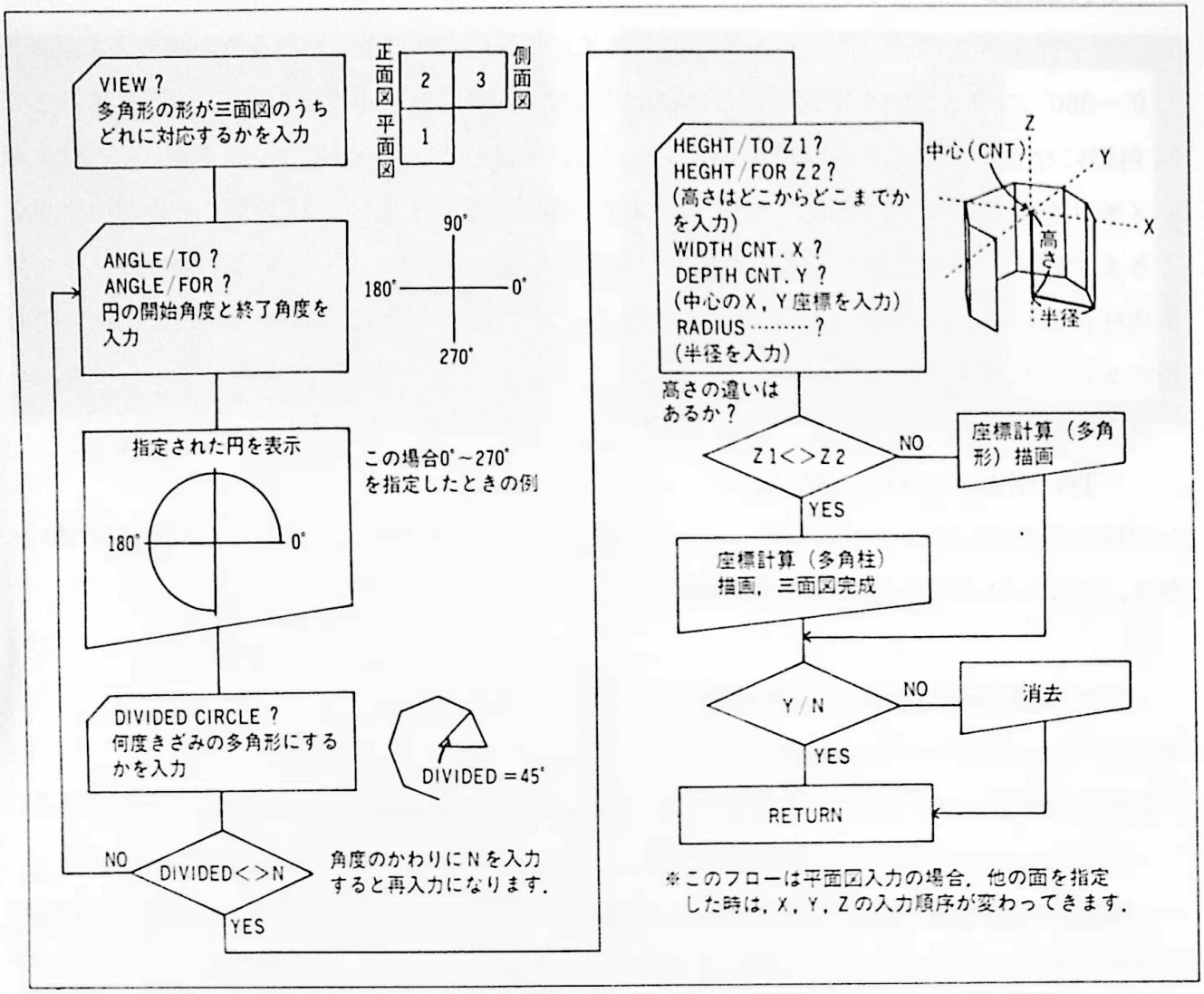
の表示となり，円の中心座標を聞いてきますので，それぞれ入力します。

最後は半径を入力します。3面図の範囲を超える大きな半径を入れると再入力です。

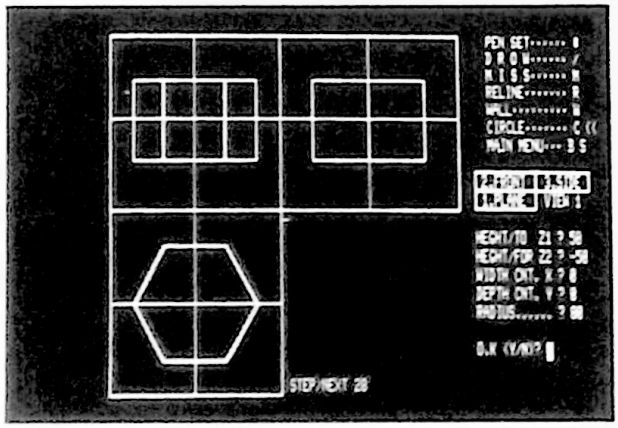
この入力を終えると，あとは自動的に円の座標が計算され，3面図に円が描画されます。

WALL，CIRCLEともに，平面図入力を例にあげ説明しましたが，正面図，側面図の場合も同様に行なってください。ただし，X，Y，Zの入力順は変わります。

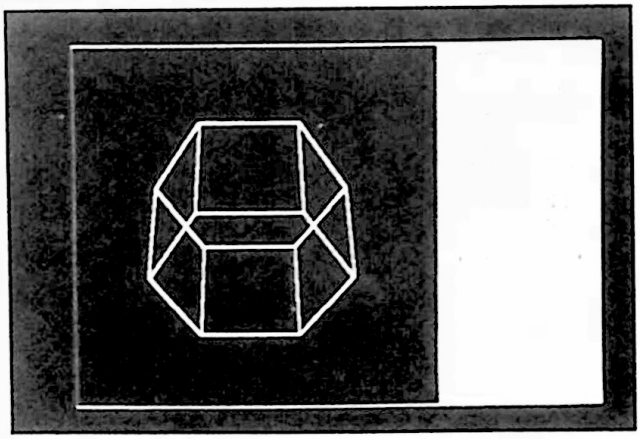
文章にすると，とてもめんどうな感じに受け取られてしまいそうですが，実際にやってみるとすぐに慣れると思います。複雑な曲面を持つモデル以外でしたら，この2つのモードでたいていのものはこなせますので，積極的に活用してください。



(図5-14　CIRCLE モードの入力手順)



(写真5-25　CIRCLE の入力例)

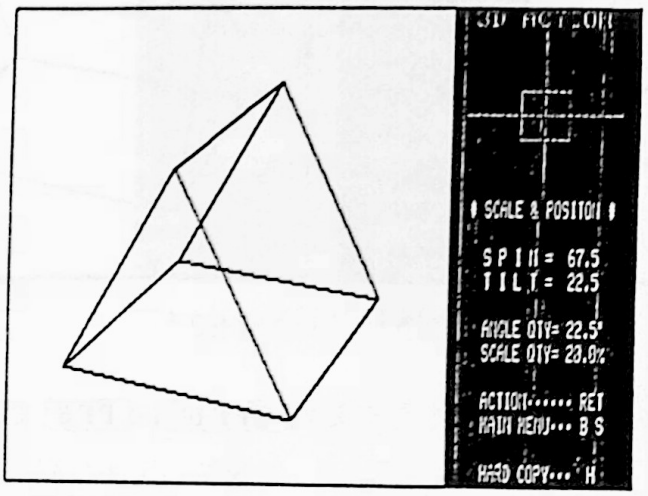


(写真5-26　CIRCLE の立体図描画例)

■ アクションモード

入力された3次元データを加工し，さまざまな角度や大きさの立体図を描かせるのがこのモードです。

メインメニューで「AC↵」と入力すると，図5-15の画面に変わります。右上の十字線の入ったワクは，現在の視線の位置と立体図の大きさを表わします。

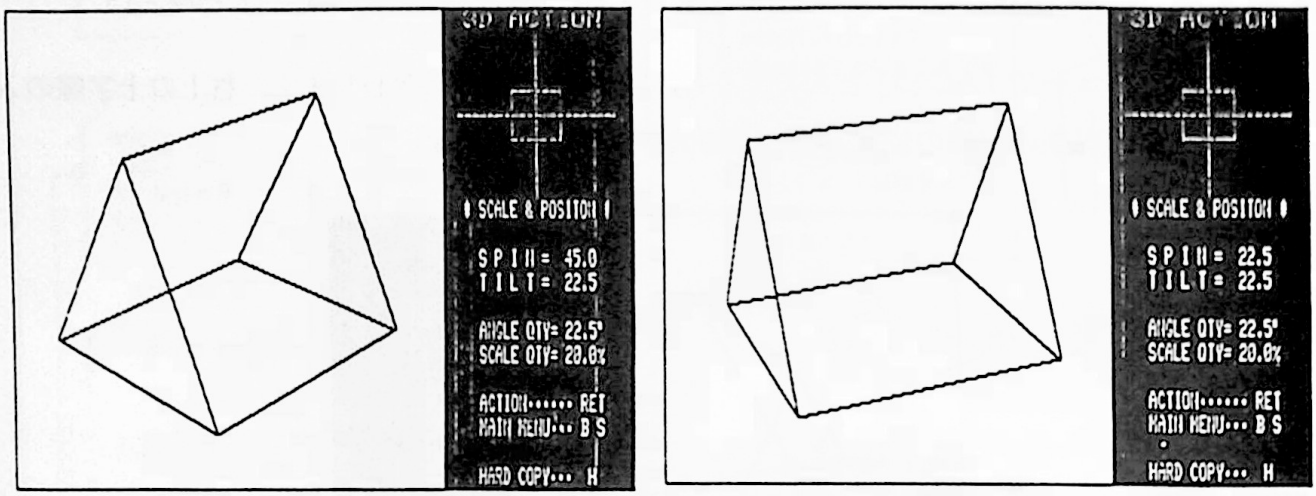


（図 5-15 アクションモード）

その下の2行は，SPIN（左右の回転角），TILT（上下の回転角）の表示です。SPINの初期値は90°で，この値は真正面から見る形になります。0°で右側，180°で左側，270°で裏側から見ることになります。TILTは0°で水平，90°で真上からとなります。

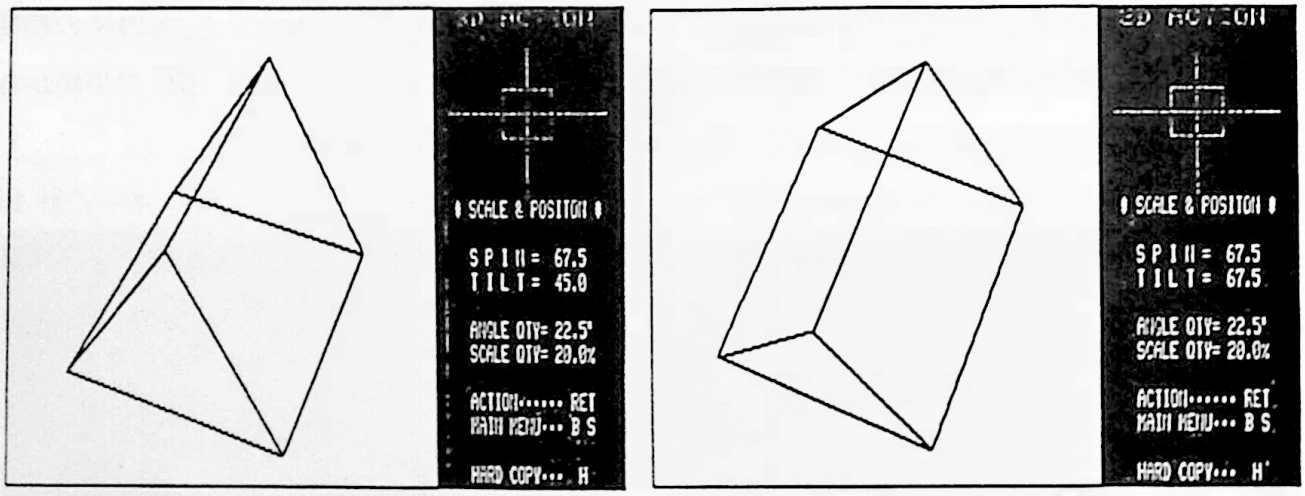
角度や大きさを変化させるためには，テンキー，カーソルキー，ファンクションキーの計16キーを使用します。これらのキーは，押した回数分だけ角度や大きさの比率が加算または減算される仕組みになっています。

4 6　SPIN（左右の回転）の角度を与えます。1回押すごとに22.5度ずつ変化しますが，PF 10，PF 5でこの単位を調整できます。



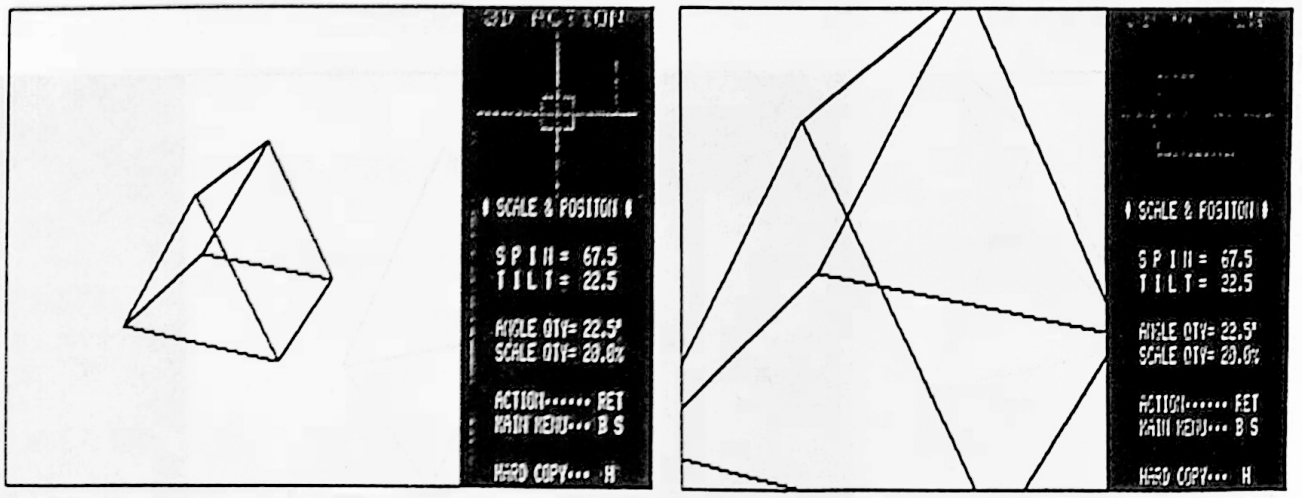
（図5-16, 17　SPINのようす）

2 8　TILT（上下の回転）の角度を与えます。PF 10，PF 5で，一度に変える角度を調整できます。



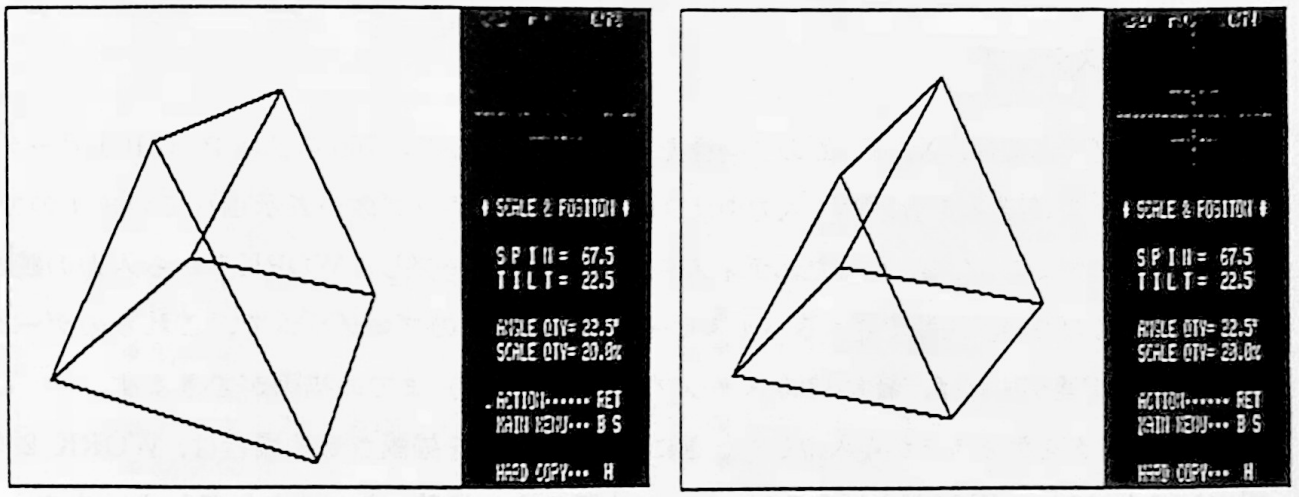
（図5-18〜19　TILTのようす）

[1] [7]　立体図の大きさが調整できます。1を押すと小さくなり，7は大きくなります。一度に変えられる大きさの比率は[PF9]，[PF4]で調整可能です。



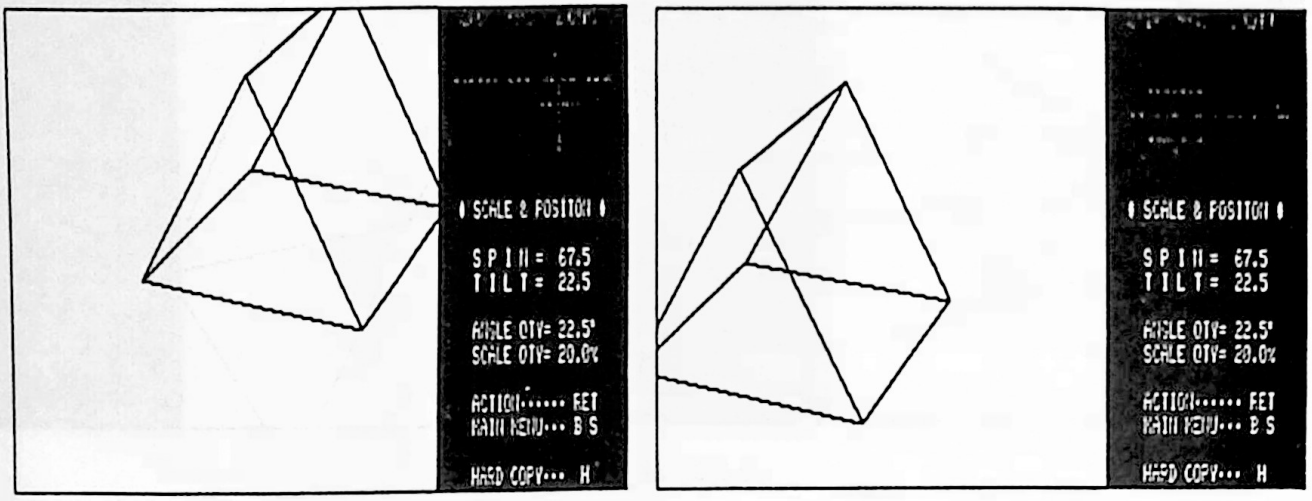
(図 5-20, 21　大きさの変化)

[3] [9]　視角を調整します。3を押すと視角が狭くなり，9で広くなります。前記の[1]，[7]と組み合わせることにより，同じモデルを異なった遠近感で見ることができますが，視覚を広げ過ぎてしまうと遠近感が誇張されすぎて，わけのわからない図になってしまうのでほどほどにしてください。



(図 5-22, 23　視角の違い)

↑↓←→　視線の位置を変えます。立体図を大きくすると当然，図の周辺は画面内におさまりきれなくなりますので，その場合，例えば右隅を見たいときに←キーを押せば，右隅の陰れた部分を中央に持ってくることができます。



(図 5-24, 25　視線の移動)

↵　前記の各キーで，描画させる角度や大きさを設定し終えたら，リターンキーで描画開始です。

←　メインメニューに戻ります。

H　ハードコピーを取ります。

● 描画ステップ

このプログラムでは，X，Y，Zの3座標を1ステップとして，700ステップ（2100データ）までの入力および描画が可能です。入力中は3面図下部にステップ数が表示されていますので，700を超えそうになったら，いったんディスクにデータをセーブし，WORK 1から入力の続きを行なってください。次節で述べるハイスピードバージョンのプログラムで，これらのデータをLOAD 2で連結すれば，最大1200ステップ（3600データ）までの描画ができます。

なお，データをディスクから入力したときに，その3面図を描画させる場合は，WORK 2を指定してください。WORK 1にすると，せっかく読み込んだデータが消されてしまいます。

● エラーの処理

700ステップ以上のデークを入力したり，Drive Not Ready等ディスク関係のエラーが発生したときは，メッセージを表示してメインメニューに戻ります。エラーの原因をつかみ，それを処理したうえで，再度メインメニューから始めてください。

● ディスクが無い場合の実行

ディスクを使う前提でプログラムの説明を続けてきましたが，カセットテープでの実行もむろん可能です。テープ使用のROMモードの場合は，ディスクベーシックが無い分フリーエリアが広くなりますので，描画ステップ数が多く確保できるメリットもあります。プログラム先頭のCLEAR文を，CLEAR，&H7FFFとし，1810行の配列を大きくすれば，さらにデークが入ります。なお，プログラム内の「DISK」の文字は「CASSETE」に置き変えてください。

◆読むだけでよくわかる

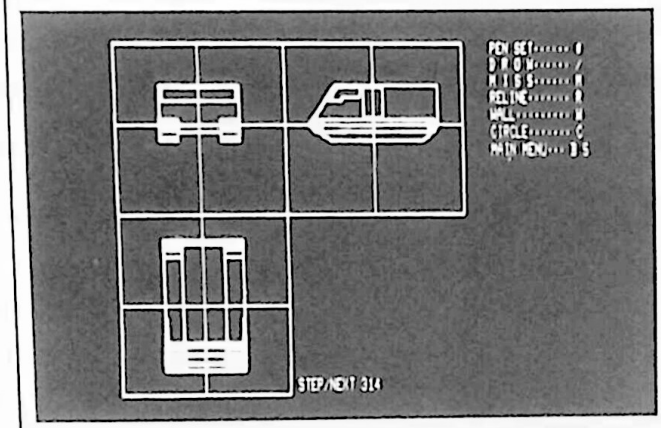
山海堂発行

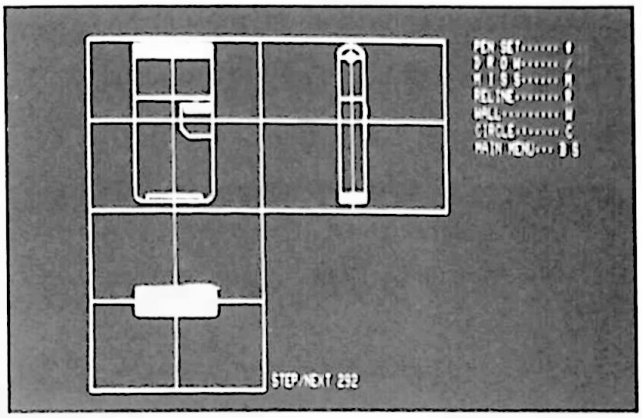
ベーシック文法入門

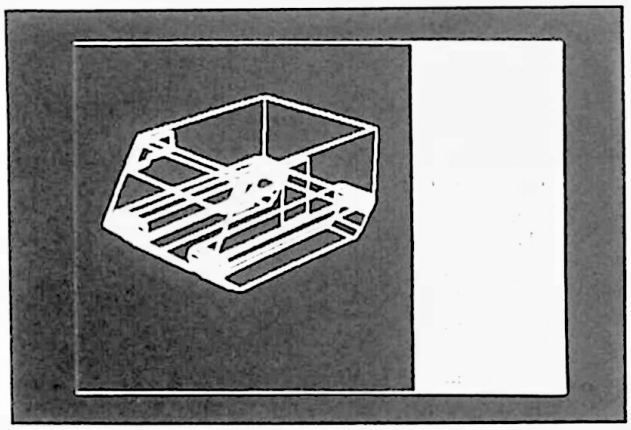
小笠原一・柳谷浩三 著／B5変形 220頁／1800円

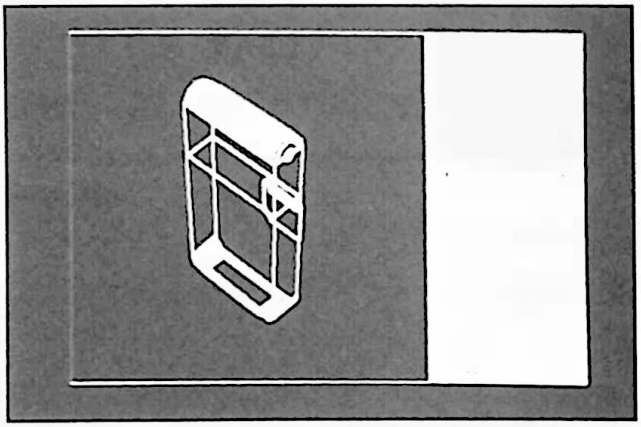
基本的な命令・よく使う命令122語を選び，意味と使い方を説明している。また，命令の機能を理解しても，一本のプログラムを作ることは大変なことです。本書では，プログラムの組み立て方や設計のポイントについても説明しております。応用として名刺管理プログラムと汎用集計プログラムの2本を収載しています。

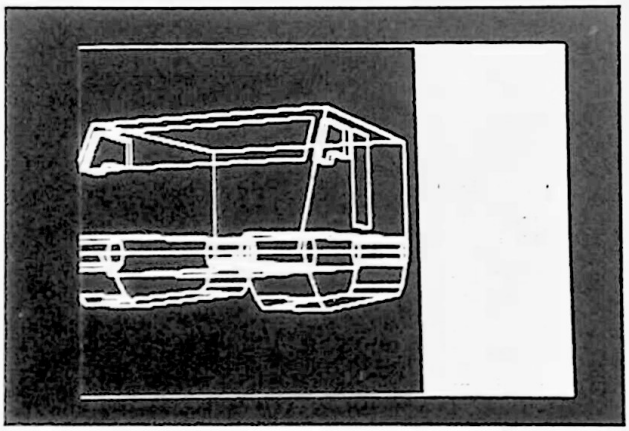
- ●BASICのまえに
- ●出力命令
- ●入力命令
- ●データの型と表現
- ●制御命令
- ●サブルーチン
- ●表示・印刷の制御
- ●ファイル処理
- ●コマンドとは
- ●応　用　編

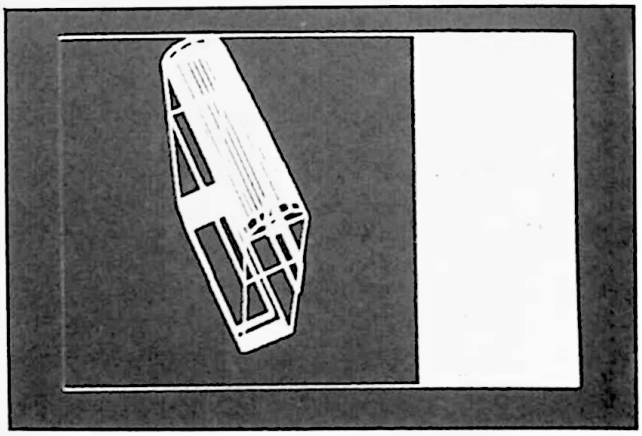












（写真 5-27～29　雪上車の描画例）

（写真 5-30～32　シェーバーの描画例）

(リスト5-2 3D WORLD Ver2)

page 1 [File Name : 3DWLD2]

```
1000 '============================
1010 '    F M   3 D   W O R L D
1020 '    ver.2     (C) S.Tajima
1030 '============================
1040 CLS:WIDTH80,20:CLEAR 300,&H6FFF:DEFSNG A-Z
1050 GOSUB 1810
1060 CLS:CLX=444:CLY=199:KA=8:KZ=.2:CX=222:CY=99
1070 ON KEY(6) GOSUB 4340:ON KEY(9) GOSUB 4440:ON KEY(4) GOSU
     B 4460
1080 ON KEY(10) GOSUB 4380:ON KEY(5) GOSUB 4410
1090 KEY(6)ON:B=1:MC=5:GOTO 1540
1100 '-----------------------------------
1110 GOSUB 2760:GOSUB 1900:CONNECT(222,0)-(222.199)-(0,199)-(
     0,0)-(444,0)-(444,99)-(0,99),3
1120 LINE(0,49)-(444,49),PSET,1:LINE(111,0)-(111,199),PSET,1
1130 LINE(0,149)-(222,149),PSET,1:LINE(333,0)-(333,99),PSET,1
     :RETURN
1140 '
1150 '************ WORKING ***************
1160 GOTO 1200
1170 CX=333:CY=149
1180 OM=1322:SC=-1.1781:SD=1.3927:D=92000!
1190 S1=SIN(SC):C1=COS(SC):S2=SIN(SD):C2=COS(SD):RETURN
1200 FOR K=7 TO 18:LOCATE 58,K:PRINT SPACE$(22);:NEXT K
1210 COLOR 4:LOCATE60,0:PRINT"PEN SET..... *"
1220 LOCATE60,1:PRINT"D R O W..... /":LOCATE60,2:PRINT"M I S
      S..... M"
1230 LOCATE 60,3:PRINT"RELINE..... R"
1240 LOCATE 60,4:PRINT"WALL........ W"
1250 LOCATE 60,5:PRINT"CIRCLE..... C"
1260 LOCATE 60,6:PRINT"MAIN MENU... B S"
1270 LOCATE 29,19:PRINT "STEP/NEXT";CO+1;:FOR I=0 TO 6:BEEP B
     :BEEP 0:D$=INKEY$
1280 LOCATE 76,I:PRINT"<<":FOR TM=0 TO 50:NEXT TM:LOCATE 76,I
     :PRINT"  "
1290 'BEEP B
1300 IF D$="*"THEN LOCATE 76,0:PRINT"<<":GOTO 1380
1310 IF D$="/"THEN LOCATE 76,1:PRINT"<<":GOTO 1380
1315 IF D$="H"THEN HARDC1
```

page 2　[File Name : 3DWLD2]

```
1320 IF D$="R"THEN LOCATE 76,3:PRINT"<<":GOTO 1380
1330 IF D$="W"THEN LOCATE 76,4:PRINT"<<":GOTO 1380
1340 IF D$="M"THEN LOCATE 76,2:PRINT"<<":GOTO 1380
1350 IF D$="C"THEN LOCATE 76,5:PRINT"<<":GOTO 1380
1360 IF D$=CHR$(&H08)THEN BEEP 0:GOTO 1540
1370 NEXT I:GOTO 1270
1380 GOSUB 4370:CO=CO+1
1390 IFD$="R"THEN CO=CO-1:GOSUB 2090:GOTO1200
1400 IF D$="M"THEN GOSUB 2140:GOTO 1200
1410 IF D$="C"THEN GOSUB 3630:CO=CO-1:GOTO1150
1420 IF D$="W"THEN GOSUB 2910:GOTO 1200
1430 GOSUB 4370
1440 LOCATE 58,8:INPUT"WIDTH.X ";B$:GOSUB 3200
1450 IF F=1 THEN 1440
1460 WX(CO)=VAL(B$):IF D$="*" THEN WX(CO)=WX(CO)+1000
1470 LOCATE 58,9:INPUT"DEPTH.Y ";B$:GOSUB 3200
1480 WY(CO)=VAL(B$):IF F=1 THEN 1470
1490 LOCATE 58,10:INPUT"HEGHT.Z ";B$:GOSUB 3200
1500 WZ(CO)=VAL(B$):IF F=1 THEN 1490
1510 IFD$="*"THEN GOSUB2020ELSE GOSUB 1920
1520 GOTO1200
1530
1540 '************ MAIN MENU **************
1550 GOSUB 1900
1560 ON ERROR GOTO 1760
1570 COLOR 7:LOCATE 60,0:PRINT"3D ACTION····· AC"
1580 LOCATE 60,1:PRINT"SAVE 3D DATA··· SA"
1590 LOCATE 60,3:PRINT"WORK 1 (NEW)··· W1"
1600 LOCATE 60,4:PRINT"WORK 2 (CONT)·· W2"
1610 LOCATE 60,6:PRINT"LOAD 1 (NEW)··· L1"
1620 LOCATE 60,7:PRINT"LOAD 2 (CONT)·· L2"
1630 LOCATE 60,9:PRINT"LLIST 3D DATA·· LD"
1635 LOCATE 60,10:PRINT"E N D ····· E"
1640 LOCATE 60,19:PRINT"STEP/NOW";CO;
1650 GOSUB 4370
1660 LOCATE 60,12:INPUT D$
1670 IF D$="AC" THEN GOSUB 2250:GOSUB 2270
1680 IF D$="SA" THEN GOSUB 4520
1690 IF D$="W2" THEN GOSUB 1170:GOSUB 2090:GOSUB 1200
```

page 3　[File Name : 3DWLD2]

```
1700 IF D$="W1" THEN GOSUB 1170:GOSUB 1830
1710 IF D$="L1" THEN GOSUB 4620
1720 IF D$="L2" THEN GOSUB 4640
1730 IF D$="LD" THEN GOSUB 4250
1735 IF D$="E" THEN FOR KY=1 TO 6:KEY(KY) OFF:NEXT:END
1740 GOTO 1660
1750 '-------------ERROR
1760 CLS:BEEP:IF ERR=9 THEN PRINT"ｺﾚｲｼﾞｮｳ ﾃﾞｰﾀﾊ ﾊｲﾘﾏｾﾝ";:CO=C
     O-1:GOTO 1800
1770 IF ERR=72 THEN PRINT"ﾃﾞｨｽｸ ｶﾞ OFF ﾃﾞｽ";:GOTO 1800
1780 IF ERR=63 THEN PRINT"ﾌｧｲﾙｶﾞ ﾐﾂｶﾘﾏｾﾝ";:GOTO 1800
1790 IF ERR=64 THEN PRINT"ﾌｧｲﾙﾉ ﾆｼﾞｭｳﾄｳﾛｸ ﾃﾞｽ";:GOTO 1800
1800 PRINT"  ﾒﾆｭｰ ﾆ ﾓﾄﾞﾘﾏｽ":RESUME 1540
1810 DIM WX(700),WY(700),WZ(700):RETURN
1820 '-------------Y/N WORK 1
1830 IF CO<=1 THEN 1880
1840 LOCATE 60,12:PRINT"Are you sure ?(Y/N)"
1850 C$=INKEY$:IF C$="Y" THEN 1880
1860 IF C$="N" THEN LOCATE 60,12:PRINT SPACE$(22):GOTO 1890
1870 GOTO 1850
1880 ERASE WX,WY,WZ:GOSUB 1810:CO=0:GOSUB 1110:GOSUB 1200:GOT
     O 1890
1890 RETURN
1900 LINE(445,0)-(639,199),PSET,0,BF:RETURN
1910 '
1920 '********** DROW **********
1930 IF CO=0 THEN CO=1
1940 WX=WX(CO-1):WY=WY(CO-1):WZ=WZ(CO-1):IF WX>=800 THEN WX=W
     X-1000
1950 VX=WX(CO):IF VX>=800 THEN VX=VX-1000
1960 LINE(WX+111,-(WZ*.45)+49)-(VX+111,-(WZ(CO)*.45)+49),PSET
     ,MC
1970 LINE(WX+111,-(WY*.45)+149)-(VX+111,-(WY(CO)*.45)+149),PS
     ET,MC
1980 LINE(WY+333,-(WZ*.45)+49)-(WY(CO)+333,-(WZ(CO)*.45)+49),
     PSET,MC
1990 LOCATE 38,19:PRINTCO;
2000 RETURN
2010
```

page 4　[File Name : 3DWLD2]

```
2020 '********** POINT ***********
2030 PSET(WX(CO)-1000+111,-(WZ(CO)*.45)+49,MC)
2040 PSET(WX(CO)-1000+111,-(WY(CO)*.45)+149,MC)
2050 PSET(WY(CO)+333,-(WZ(CO)*.45)+49,MC)
2060 LOCATE 38,19:PRINTCO;
2070 RETURN
2080 '
2090 '********** RELINE ********
2100 IF CO<=1 THEN GOSUB 1110:CO=0:RETURN
2110 JJ=CO:GOSUB1110:FOR CO=1 TO JJ:IF WX(CO)>=800 THEN GOSUB
      2020:GOTO 2130
2120 GOSUB 1930
2130 NEXT CO:CO=CO-1:RETURN
2140 '******** MISS ***********
2150 IF CO= 0 THEN 2190 ELSE CO=CO-1
2160 MC=0:IF WX(CO)>=800 THEN GOSUB 2020:GOTO 2180
2170 GOSUB 1930
2180 CO=CO-1:MC=5:IF CO=-1 THEN CO=0
2190 RETURN
2200 '-----------------
2210 ZC=OM-X*C1*S2-Y*S1*S2-Z*C2:QX=(D*(-X*S1+Y*C1)/ZC)+CX:QY=
     CY-.4495*(D*(-X*C1*C2-Y*S1*C2+Z*S2)/ZC):RETURN
2220 '
2230 '******** ACTION *********
2240 GOSUB 4370:GOTO 2270
2250 CX=222:CY=99:OM=10:D=1500:SC=-1.5708:SD=1.5708:S1=SIN(SC
     ):C1=COS(SC)
2260 KEY(10) ON:KEY(5) ON:KEY(9)ON:KEY(4)ON:S2=SIN(SD):C2=COS
     (SD):RETURN
2270 GOSUB 1900:LINE(445,0)-(639,199),PSET,4,BF:COLOR12:SYMBO
     L(467,1),"3D ACTION",2,1,1:SYMBOL(465,1),"3D ACTION",2,1,
     3
2280 GOSUB 4430:GOSUB 4480:LINE(460,96)-(620,120),PSET,2,B
2290 LOCATE 58,8:PRINT"* SCALE & POSITON *":LINE(460,78)-(620
     ,88),PSET,1,B
2300 GOSUB 2520:LOCATE 60,16:PRINT"ACTION····· RET":LOCATE 6
     0,17:PRINT"MAIN MENU··· B S":LOCATE 60,19:PRINT"HARD COPY
     ···  H";
2310 A$=INKEY$:IFA$="3"THEN OM=OM+OM*KZ:GOSUB 2520
```

page 5 [File Name : 3DWLD2]

```
2320 IF A$="9"THEN OM=OM-OM*KZ:GOSUB 2520
2330 IF A$="7"THEN D=D+D*KZ:GOSUB 2520
2340 IF A$="1"THEN D=D-D*KZ:GOSUB 2520
2350 IF A$="6"THEN SC=SC+3.14159/KA:S1=SIN(SC):C1=COS(SC):GOT
     O 2480
2360 IF A$="4"THEN SC=SC-3.14159/KA:S1=SIN(SC):C1=COS(SC):GOT
     O 2480
2370 IF A$="8"THEN SD=SD+3.14159/KA:S2=SIN(SD):C2=COS(SD):GOT
     O 2480
2380 IF A$="2"THEN SD=SD-3.14159/KA:S2=SIN(SD):C2=COS(SD):GOT
     O 2480
2390 IF A$=CHR$(&H1C) THEN CX=CX+40:GOSUB 2520:GOTO 2500
2400 IF A$=CHR$(&H1D) THEN CX=CX-40:GOSUB 2520:GOTO 2500
2410 IF A$=CHR$(&H1E) THEN CY=CY-20:GOSUB 2520:GOTO 2500
2420 IF A$=CHR$(&H1F) THEN CY=CY+20:GOSUB 2520:GOTO 2500
2430 IF A$="H" THEN HARDC1
2440 IF A$=CHR$(13) THEN GOSUB 2650:GOTO 2310
2450 IF A$=CHR$(&H08) THEN KEY(10)OFF:KEY(5)OFF:KEY(9)OFF:KEY
     (4)OFF:COLOR 4:GOTO 1540
2460 LOCATE 60,13:PRINT"ANGLE QTY=";:PRINT USING"###.#";AN;:P
     RINT"°  ";
2470 LOCATE 60,14:PRINT"SCALE QTY=";:PRINT USING"###.#";SL;:P
     RINT"%";
2480 LOCATE 60,10:PRINT"S P I N =";:PRINTUSING"####.#";(SC/(3
     .14159/180))*-1
2490 LOCATE 60,11:PRINT"T I L T =";:PRINTUSING"####.#";(SD/(3
     .14159/180)-90)*-1
2500 GOTO2310
2510 '----------SENTER.SCALE
2520 LINE(460,10)-(620,78),PSET,1,BF
2530 CPX=160/444*CX+460:IF CPX<460 THEN CPX=460
2540 IF CPX>620 THEN CPX=620
2550 CPY=68/199*CY+10:IF CPY<10 THEN CPY=10
2560 IF CPY>78 THEN CPY=78
2570 LINE(CPX,10)-(CPX,78),PSET,5:LINE(460,CPY)-(620,CPY),PSE
     T,5
2580 PP=(D/1500)*(10/OM)
2590 FOR CPS=2 TO 5 STEP 3
2600 PX1=CPX-CPS*5*PP:IF PX1<460 THEN PX1=460
```

page 6　[File Name : 3DWLD2]

```
2610 PX2=CPX+CPS*5*PP:IF PX2>620 THEN PX2=620
2620 PY1=CPY-CPS*2*PP:IF PY1<10 THEN PY1=10
2630 PY2=CPY+CPS*2*PP:IF PY2>78 THEN PY2=78
2640 LINE(PX1,PY1)-(PX2,PY2),PSET,CPS,B:NEXT:RETURN
2650 '********** view work **********
2660 GOSUB 2760:LINE(0,0)-(444,199),PSET,4,B:FOR I=1 TO CO
2670 IF WX(I)=>800 THEN 2750
2680 X=WX(I)/100:Y=WY(I)/100:Z=WZ(I)/100:GOSUB 2210:X1=QX:Y1=
     QY
2690 IF WX(I-1)>800 THEN X=(WX(I-1)-1000)/100 ELSE X=WX(I-1)/
     100
2700 Y=WY(I-1)/100:Z=WZ(I-1)/100:GOSUB 2210:X0=QX:Y0=QY
2710 IF (X0>0)+(X0<444)+(Y0>0)+(Y0<199)+(X1>0)+(X1<444)+(Y1>0
     )+(Y1<199)=-8 THEN2740
2720 IF (X0<0 AND X1<0) OR (X0>444 AND X1>444) OR (Y0<0 AND Y
     1<0) OR (Y0>199 AND Y1>199) THEN BEEP B:BEEP 0:GOTO 2750
2730 GOSUB 2790:SWAP X0,X1:SWAP Y0,Y1:GOSUB 2790
2740 LINE(X0,Y0)-(X1,Y1),PSET,7
2750 NEXT I:GOSUB 4370:RETURN
2760 LINE(0,0)-(444,199),PSET,0,BF:RETURN
2770
2780   ********** CLIPPING **********
2790 IF X1<0 THEN XX=0:GOTO 2820
2800 IF X1>CLX THEN XX=CLX:GOTO 2820
2810 XX=X1
2820 IF X1=X0 THEN YY=Y1:GOTO2860
2830 YY=Y1-(Y1-Y0)*(X1-XX)/(X1-X0)
2840 IF YY<0 THEN YY=0:GOTO 2870
2850 IF YY>CLY THEN YY=CLY:GOTO2870
2860 X1=XX:Y1=YY:GOTO 2890
2870 IF Y1=Y0 THEN XX=X1:GOTO 2860
2880 XX=X1-(X1-X0)*(Y1-YY)/(Y1-Y0):GOTO 2860
2890 RETURN
2900 '
2910 '********** WALL **************
2920 GOSUB 4160:IF C$="N" THEN RETURN
2930 IF VAL(C$)>4 OR VAL(C$)<1 THEN 2920
2940 ON VAL(C$)GOTO 2960,3230,3430
2950 GOTO2920
```

page 7 [File Name : 3DWLD2]

```
2960 LOCATE 58,11:INPUT"HEGHT/TO  Z1 ";B$:GOSUB 3200:TZ=VAL(B
     $):IF F=1THEN 2960
2970 LOCATE 58,12:INPUT"HEGHT/FOR Z2 ";B$:GOSUB 3200:IF F=1 T
     HEN 2970
2980 FZ=VAL(B$):BA=CO:BC=0:GOSUB 3020:WX(CO)=WX(CO)+1000:GOSU
     B 2020
2990 CO=CO+1:LOCATE 58,14:PRINT SPC(20):LOCATE 58,15:PRINTSPC
     (20):GOSUB 3020:IF B$="E" THEN CO=CO-1:IF TZ<>FZ THEN GOS
     UB 3140:RETURN
3000 IF B$="E" THEN RETURN
3010 GOSUB 1920:GOTO 2990
3020 LOCATE 58,14:PRINT "WIDTH... X ";:INPUT B$:GOSUB 3200:IF
      F=1 THEN 3020
3030 IF B$="E" THEN 3080
3040 IF B$="M" AND CO>BA THEN GOSUB 2140:CO=CO+1:GOTO 3020
3050 WX(CO)=VAL(B$)
3060 LOCATE 58,15:PRINT "DEPTH... Y ";:INPUT B$:GOSUB 3200:IF
      F=1 THEN 3060
3070 WY(CO)=VAL(B$):WZ(CO)=TZ
3080 RETURN
3090 '---------------Y/N RELINE
3100 LOCATE 58,17:INPUT"O.K (Y/N)";C$:GOSUB 4370:IF C$="Y" TH
     EN RETURN
3110 :MC=0:FOR CO=BA TO BF:IF WX(CO)<800 THEN GOSUB 1930
3120 NEXT CO:CO=BA-1:MC=5:RETURN
3130 '--------------MAKE DATA/WALL
3140 WX(BA)=WX(BA)-1000:GOSUB 4500:BC=0:BS=CO+1:BE=BS+CO-BA:B
     M=BS-BA:FOR CO=BS TO BE:WX(CO)=WX(CO-BM):WY(CO)=WY(CO-BM)
     :WZ(CO)=FZ
3150 BD=CO+BM+BC:BF=BD+1:WX(BD)=WX(CO):WY(BD)=WY(CO):WX(BD)=W
     X(BD)+1000
3160 WZ(BD)=TZ:WZ(BF)=FZ:WX(BF)=WX(CO):WY(BF)=WY(CO):BC=BC+1:
     NEXT CO:WX(BS)=WX(BS)+1000:WX(BA)=WX(BA)+1000
3170 FOR CO=BS TO BF:IFWX(CO)<800 THEN GOSUB 1930
3180 NEXT CO:CO=BF:GOSUB 3100:RETURN
3190 '--------------FLUG.-110<B$>+110
3200 GOSUB 4370:F=((B$="")+(VAL(B$)<-110)+(VAL(B$)>110))*-1:I
     F F=1 THEN BEEP
3210 RETURN
```

```
page 8    [ File Name : 3DWLD2 ]
3220 '--------------FRONT VIEW
3230 LOCATE 58,11:INPUT"DEPTH/TO  Y1 ";B$:GOSUB 3200:TY=VAL(B
     $):IF F=1THEN 3230
3240 LOCATE 58,12:INPUT"DEPTH/FOR Y2 ";B$:GOSUB 3200:IF F=1 T
     HEN 3240
3250 FY=VAL(B$):BA=CO:BC=0:GOSUB 3290:WX(CO)=WX(CO)+1000:GOSU
     B 2020
3260 CO=CO+1:LOCATE 58,14:PRINT SPC(20):LOCATE 58,15:PRINT SP
     C(20):GOSUB 3290:IF B$="E" THEN CO=CO-1:IF TY<>FY THEN GO
     SUB 3370:RETURN
3270 IF B$="E" THEN RETURN
3280 GOSUB 1920:GOTO 3260
3290 LOCATE 58,14:PRINT "WIDTH... X ";:INPUT B$:GOSUB 3200:IF
      F=1 THEN 3290
3300 IF B$="E" THEN GOTO 3350
3310 IF B$="M" AND CO>BA THEN GOSUB 2140:CO=CO+1:GOTO 3290
3320 WX(CO)=VAL(B$)
3330 LOCATE 58,15:PRINT "HEGHT... Z ";:INPUT B$:GOSUB 3200:IF
      F=1 THEN 3330
3340 WZ(CO)=VAL(B$):WY(CO)=TY
3350 RETURN
3360 '----------------MAKE DATA
3370 WX(BA)=WX(BA)-1000:GOSUB 4500:BC=0:BS=CO+1:BE=BS+CO-BA:B
     M=BS-BA:FOR CO=BS TO BE:WX(CO)=WX(CO-BM):WZ(CO)=WZ(CO-BM)
     :WY(CO)=FY
3380 BD=CO+BM+BC:BF=BD+1:WX(BD)=WX(CO)+1000:WZ(BD)=WZ(CO)
3390 WY(BD)=TY:WY(BF)=FY:WX(BF)=WX(CO):WZ(BF)=WZ(CO):BC=BC+1:
     NEXT CO:WX(BS)=WX(BS)+1000:WX(BA)=WX(BA)+1000
3400 FOR CO=BS TO BF:IFWX(CO)<800 THEN GOSUB 1930
3410 NEXT CO:CO=BF:GOSUB 3100:RETURN
3420 '--------------SIDE VIEW
3430 LOCATE 58,11:INPUT"WIDTH/TO  X1 ";B$:GOSUB 3200:TX=VAL(B
     $):IF F=1THEN 3430
3440 LOCATE 58,12:INPUT"WIDTH/FOR X2 ":B$:GOSUB 3200:IF F=1 T
     HEN 3440
3450 FX=VAL(B$):BA=CO:BC=0:GOSUB 3490:WX(CO)=FX+1000:GOSUB 20
     20
3460 CO=CO+1:LOCATE 58,14:PRINT SPC(20):LOCATE 58,15:PRINT SP
     C(20):GOSUB 3490:IF B$="E" THEN CO=CO-1:IF TX<>FX THEN GO
```

page 9 [File Name : 3DWLD2]

```
    SUB 3570:RETURN
3470 IF B$="E" THEN RETURN
3480 GOSUB 1920:GOTO 3460
3490 LOCATE 58,14:PRINT "DEPTH... Y ";:INPUT B$:GOSUB 3200:IF
     F=1 THEN 3490
3500 IF B$="E" THEN 3550
3510 IF B$="M" AND CO>BA THEN GOSUB 2140:CO=CO+1:GOTO 3490
3520 WY(CO)=VAL(B$)
3530 LOCATE 58,15:PRINT "HEGHT... Z ";:INPUT B$:GOSUB 3200:IF
     F=1 THEN 3530
3540 WZ(CO)=VAL(B$):WX(CO)=FX
3550 RETURN
3560 '--------------MAKE DATA
3570 WX(BA)=WX(BA)-1000:GOSUB 4500:BC=0:BS=CO+1:BE=BS+CO-BA:B
    M=BS-BA:FOR CO=BS TO BE:WY(CO)=WY(CO-BM):WZ(CO)=WZ(CO-BM)
    :WX(CO)=TX
3580 BD=CO+BM+BC:BF=BD+1:WY(BD)=WY(CO):WZ(BD)=WZ(CO)
3590 WX(BD)=TX+1000:WX(BF)=FX:WY(BF)=WY(CO):WZ(BF)=WZ(CO):BC=
    BC+1:NEXT CO:WX(BS)=WX(BS)+1000:WX(BA)=WX(BA)+1000
3600 FOR CO=BS TO BF:IFWX(CO)<800 THEN GOSUB 1930
3610 NEXT CO:CO=BF:GOSUB 3100:RETURN
3620 '
3630 '********** CIRCLE *********
3640 GOSUB 4160:IF C$="N" THEN RETURN
3650 IF VAL(C$)>=4 OR VAL(C$)<1 THEN 3640
3660 LINE(500,175)-(600,175),PSET,1:LINE(550,152)-(550,197),P
    SET,1:LOCATE 58,17:PRINT"180°";:LOCATE 76,17:PRINT"0°";
3670 LOCATE 58,12:INPUT"ANGLE/TO..... ";B$:AT=VAL(B$):GOSUB
    4370:LOCATE 58,13:INPUT"ANGLE/FOR..... ";B$:AF=VAL(B$):GO
    SUB 4370:IF AT>AF OR AF>720 OR AT=AF THEN BEEP:GOTO 3670
3680 ATC=AT:IF ATC>360 THEN ATC=AT-360
3690 AFC=AF:IF AFC>360 THEN AFC=AF-360
3700 CIRCLE @(550,175),50,3,,1-AFC/360,1-ATC/360
3710 LOCATE 58,14:INPUT"DIVIDED CIRCLE ";B$
3720 IF B$="N" THEN GOSUB 4760:GOTO 3660
3730 IF VAL(B$)<1THEN 3710
3740 DC=VAL(B$):GOSUB 4370:IF DC=0 OR (AF-AT)*10 MOD DC*10 <>
    0 THEN BEEP:GOTO 3710
3750 FOR K=12 TO 19:LOCATE 58,K:PRINTSPACE$(21);:NEXT
```

page 10　[File Name : 3DWLD2]

```
3760 ON VAL(C$)GOTO 3770,3910,4040
3770 LOCATE 28,10:PRINT CHR$(&H95)
3780 LOCATE 58,11:INPUT"HEGHT/TO  Z1 ";B$:GOSUB 3200:TZ=VAL(B
     $):IF F=1THEN 3780
3790 LOCATE 58,12:INPUT"HEGHT/FOR Z2 ";B$:GOSUB 3200:FZ=VAL(B
     $):IF F=1THEN 3790
3800 LOCATE 58,13:INPUT"WIDTH CNT. X ";B$:GOSUB 3200:X=VAL(B$
     ):IF F=1 THEN 3800
3810 LOCATE 58,14:INPUT"DEPTH CNT. Y ";B$:GOSUB 3200:Y=VAL(B$
     ):IF F=1 THEN 3810
3820 LOCATE 58,15:INPUT"RADIUS..... ";B$:GOSUB 4370:RA=VAL(B
     $):IF ABS(RA)+ABS(X)>110 OR ABS(RA)+ABS(Y)>110 THEN 3820
3830 '--------------MAKE DATA/CIRCLE
3840 J=BA:FOR I=AT TO AF STEP DC:WX(J)=X+RA*COS(I*3.14159/180
     ):IF WX(J)>-.01 AND WX(J) <0 THEN WX(J)=0
3850 WY(J)=Y+RA*SIN(I*3.14159/180):IF WY(J)>-.01 AND WY(J)<0
     THEN WY(J)=0
3860 WZ(J)=TZ:J=J+1:NEXT I:WX(BA)=WX(BA)+1000::J=J-1
3870 FOR CO=BA+1 TO J:GOSUB 1930:NEXT CO:IF TZ<>FZ THEN CO=CO
     -1:GOSUB 3140:CO=CO+1:GOTO 3890
3880 BF=CO:BA=BA+1:GOSUB 3100:GOTO 3890
3890 LOCATE 28,10:PRINT" ":RETURN
3900 '--------------FRONT VIEW
3910 LOCATE 28,10:PRINT CHR$(&HEF)
3920 LOCATE 58,11:INPUT"DEPTH/TO  Y1 ";B$:GOSUB 3200:TY=VAL(B
     $):IF F=1 THEN 3920
3930 LOCATE 58,12:INPUT"DEPTH/FOR Y2 ";B$:GOSUB 3200:FY=VAL(B
     $):IF F=1 THEN 3930
3940 LOCATE 58,13:INPUT"WIDTH CNT. X ";B$:GOSUB 3200:X=VAL(B$
     ):IF F=1 THEN 3940
3950 LOCATE 58,14:INPUT"HEGHT CNT. Z ";B$:GOSUB 3200:Z=VAL(B$
     ):IF F=1 THEN 3950
3960 LOCATE 58,15:INPUT"RADIUS..... ";B$:GOSUB 4370:RA=VAL(B
     $):IF ABS(RA)+ABS(X)>110 OR ABS(RA)+ABS(Z)>110 THEN 3960
3970 '--------------MAKE DATA
3980 J=BA:FOR I=AT TO AF STEP DC:WX(J)=X+RA*COS(I*3.14159/180
     ):IF WX(J)>-.01 AND WX(J) <0 THEN WX(J)=0
3990 WZ(J)=Z+RA*SIN(I*3.14159/180):IF WZ(J)>-.01 AND WZ(J)<0
     THEN WZ(J)=0
```

page 11 [File Name : 3DWLD2]

```
4000 WY(J)=TY:J=J+1:NEXT I:WX(BA)=WX(BA)+1000:J=J-1
4010 FOR CO=BA+1 TO J:GOSUB 1930:NEXT CO:IF TY<>FY THEN CO=CO
     -1:GOSUB 3370:CO=CO+1:GOTO 3890
4020 GOTO 3880
4030 '--------------SIDE VIEW
4040 LOCATE 28,10:PRINT CHR$(&H96)
4050 LOCATE 58,11:INPUT"WIDTH/TO  X1 ";B$:GOSUB 3200:TX=VAL(B
     $):IF F=1THEN 4050
4060 LOCATE 58,12:INPUT"WIDTH/FOR X2 ";B$:GOSUB 3200:FX=VAL(B
     $):IF F=1THEN 4060
4070 LOCATE 58,13:INPUT"DEPTH CNT. Y ";B$:GOSUB 3200:Y=VAL(B$
     ):IF F=1 THEN 4070
4080 LOCATE 58,14:INPUT"HEGHT CNT. Z ";B$:GOSUB 3200:Z=VAL(B$
     ):IF F=1 THEN 4080
4090 LOCATE 58,15:INPUT"RADIUS..... ";B$:GOSUB 4370:RA=VAL(B
     $):IF ABS(RA)+ABS(Y)>110 OR ABS(RA)+ABS(Z)>110 THEN 4090
4100 '--------------MAKE DATA
4110 J=BA:FOR I=AT TO AF STEP DC:WY(J)=Y+RA*COS(I*3.14159/180
     )
4120 WZ(J)=Z+RA*SIN(I*3.14159/180):WX(J)=FX:J=J+1:NEXT I:WX(B
     A)=WX(BA)+1000:J=J-1
4130 FOR CO=BA+1 TO J:GOSUB 1930:NEXT CO:IF TX<>FX THEN CO=CO
     -1:GOSUB 3570:CO=CO+1:GOTO 3890
4140 GOTO 3880
4150 '--------------VIEW/1.2.3
4160 LOCATE 59,8:PRINT"2.FRONT   3.SIDE":LOCATE 59,9:PRINT"1.
     PLANE":BA=CO
4170 LOCATE 59,9:PRINT"1.PLANE":BA=CO
4180 LINE(464,78)-(610,88),PSET,3,B:LINE(464,78)-(540,98),PSE
     T,3,B
4190 LOCATE 69,9:PRINT"VIEW ?":C$=INPUT$(1):GOSUB 4370:IF C$
     <>""THEN 4200 ELSE 4190
4200 LOCATE 74,9:PRINT C$:RETURN
4210 '
4220 PRINT "/";PRINTUSING"######.#";WX(DL);
4230 '
4240 '
4250 '********** DATA/LLIST ************
4260 LPRINT"   No.        ( X )   ( Y )   ( Z )"
```

page 12　[File Name : 3DWLD2]

```
4270 FOR DL=1TO CO:LPRINTUSING"####";DL;
4280 IF WX(DL)>800 THEN LPRINT"    [*]";:LPRINTUSING"#######.#
     ";WX(DL)-1000;:GOTO 4300
4290 LPRINT"    [/]";:LPRINTUSING"######.#";WX(DL);
4300 LPRINTUSING"######.#";WY(DL);WZ(DL):NEXT
4310 RETURN
4320 '
4330 '******** BEEP SWICH ********
4340 B=B+1:IF B=2 THEN B=0
4350 RETURN
4360 '--------------------
4370 BEEP B:FOR BT=0 TO 50:NEXT:BEEP 0:RETURN
4380 '****** ANGLE. SCALE. ********
4390 KA=KA/2:IF KA<.1 THEN KA=1
4400 GOSUB 4430:RETURN
4410 KA=KA*2:IF KA>128 THEN KA=128
4420 GOSUB 4430:RETURN
4430 AN=3.14159/KA/(3.14159/180):RETURN
4440 KZ=KZ*2:IF KZ>1 THEN KZ=.8
4450 GOSUB 4480:RETURN
4460 KZ=KZ/2:IF KZ<.1 THEN KZ=.1
4470 GOSUB 4480:RETURN
4480 SL=KZ*100:RETURN
4490 '------------------
4500 LOCATE 58,17:PRINT"BUSY !!";:RETURN
4510 '
4520 '********** 3D DATA SAVE *************
4530 COLOR 5:LOCATE 60,18:INPUT"FILE NAME";TITLE$:IF LEN(TITL
     E$)>8 THEN BEEP:GOTO 4530
4540 TITLE$=TITLE$+SPACE$(8-LEN(TITLE$)):LOCATE 60,14:PRINT"R
     EADY D E S K !":LOCATE 60,15
4550 LOCATE 60,17:PRINT"HIT RETURN KEY":COLOR 2
4560 BEEP 0:C$=INKEY$:LOCATE 60,15:PRINT"( R E C O R D )"
4570 FOR BB=0 TO 50:NEXT:BEEP 1:IF C$ <> CHR$(13) THEN LOCATE
      60,15:PRINT"                ":GOTO 4560
4580 LOCATE 60,17:PRINT"                    ":BEEP 0:OPEN"O",#1,TIT
     LE$
4590 PRINT#1,TITLE$:PRINT#1,CO:FOR S=1 TO CO:PRINT#1,WX(S)
4600 PRINT#1,WY(S):PRINT#1,WZ(S):LOCATE 75,17:PRINT S;:NEXT S
```

```
page 13    [ File Name : 3DWLD2 ]
4610 CLOSE #1:GOSUB 4760:RETURN
4620 '********** LOAD 1 **********
4630 CO=0:SI=CO:GOSUB 4670:RETURN
4640 '********** LOAD 2 **********
4650 SI=CO:GOSUB 4670:RETURN
4660 '------------------------------
4670 COLOR 5:LOCATE 60,14:PRINT"READY D E S K !":LOCATE 60,15
     :LOCATE 60,18:INPUT"FILE NAME";TITLE$:IF LEN(TITLE$)>8 TH
     EN BEEP:GOTO 4670
4680 TITLE$=TITLE$+SPACE$(8-LEN(TITLE$)):LOCATE 60,17:PRINT"H
     IT RETURN KEY"
4690 BEEP 0:C$=INKEY$:LOCATE 60,15:PRINT"(  P  L  A  Y  )"
4700 FOR BB=0 TO 50:NEXT:BEEP 1:IF C$ <> CHR$(13) THEN LOCATE
      60,15:PRINT"                    ":GOTO 4690
4710 LOCATE 60,17:PRINT"                      ":BEEP 0:OPEN"I",#1,TIT
     LE$
4720 INPUT#1,TITLE$:LOCATE 60,17:PRINT "Found:";TITLE$:INPUT#
     1,CT:CO=CO+CT:FOR L=SI+1 TO CO:INPUT#1,WX(L)
4730 INPUT#1,WY(L):INPUT#1,WZ(L):LOCATE 75,17:PRINT L;:NEXT L
4740 '
4750 CLOSE #1:GOSUB 4760:RETURN
4760 LINE(445,120)-(639,199),PSET,0,BF:GOSUB 4370:RETURN
```


◆サブルーチンとジェネレータを使った

山海堂発行

F-BASIC (FM-NEW7/-77/-7)

ゲーム作りのノーハウ

ナラシノマイコンクラブ著／B5判 208頁／2300円

ゲーム作りにこそプログラミングの真髄があるという人もおります．しかし，無用の骨折りはムダそのものです．本書では数多くのサブルーチンとパターンジェネレータ，サウンドジェネレータのプログラムと，それらを使って作ったゲームを収載しております．ゲームを作ってみようという人，効率的に作ろうという人に役立つ本です．

- ゲーム・プログラムを作る前に
- プログラム・テクニックあれこれ
- HANOIの塔
- UFOとレーザー砲
- パターン・ジェネレータ
- UFOとミサイル
- 15パズル
- サウンド・ジェネレータ

3次元高速描画バージョン (3D WORLD Ver 2.1 Hispeed version)

このプログラムは前節のVer 2と併用して使います。Ver 2で作成し保存した3次元データのディスクを，このプログラムで読み込むと立体図の高速描画が可能となります。また，Ver 2に無い機能も加えてみました。

(ハイスピードバージョンの機能と特徴)

(1) マシン語による高速描画
(2) 立体図を動かすオートアニメーション
(3) プロッタへの出力
(4) 画面の保存
(5) 描画ステップ数の拡大（1200ステップ）

以上，5つの主な機能・特徴がありますが，3面図モードの入力ルーチンは省きました。できあがったデータの描画専用プログラムとして利用してください。

■ プログラムの実行

RUN直後はメニューから始まります。メニューの内容は，図5-26で示すように，

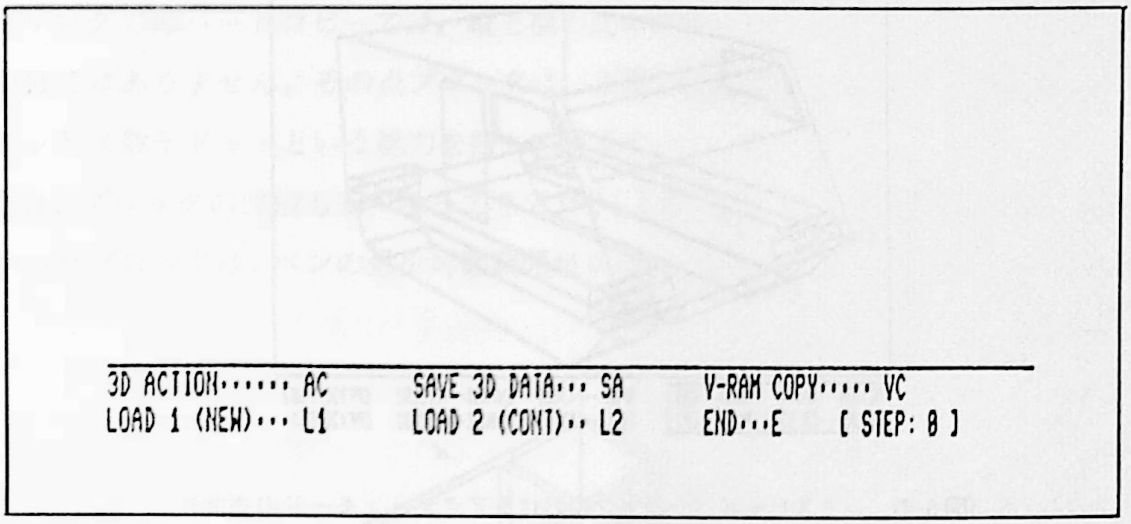
3D ACTION …… AC
SAVE 3D DATA …… SA
LOAD 1(NEW) …… L 1
LOAD 2(CONT) …… L 2
V-RAM COPY …… VC

となっています。V-RAM COPY以外はVer 2と同じですので，前節を参照してください。

ここで，いきなり「AC ↵」とアクションモードを指定しても，データが入っていませんのでモードの移行はできません。まずLOAD 1を指定し，データを読み込むことが先決です。

V-RAM COPYは，画面をそっくりディスクに書き込むことを意味し，第3章の絵を描くプログラム「スケッチボード2」で使用した画面保存と同じ動作を行ないます。

ファイル名を入力すると，画面上部から1ラインずつディスクに保存します。このディスクを，「スケッチボード2」プログラムで読み込めば画面の再生ができ，その立体図に色を塗ったり，細部の仕上げも可能となります。



(図5-26　ハイスピードバージョンのメインメニュー)

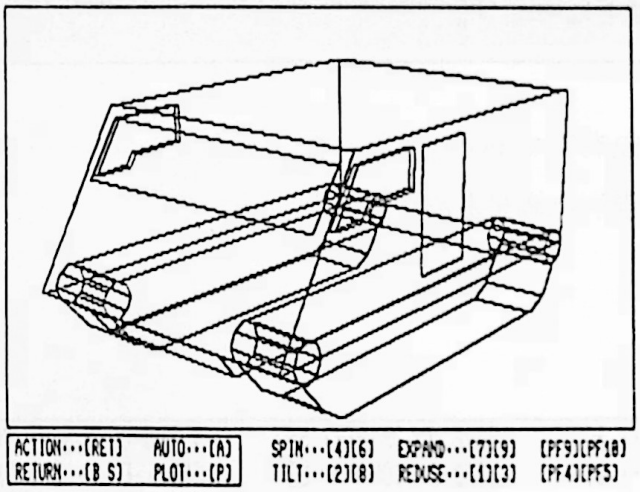
■　アクションモード

データのLOAD後，メインメニューでAC↵とキーインすると，このモードに入ります。描画ルーチンはマシン語で組みましたので，Ver 2 のベーシックだけで組んだものに比べると，その速度差は歴然とした違いがあります。ベーシックでは，ポツポツと描いていた立体図を，さすがマシン語，「ダーッ」という感じで描き上げてしまいます。しかし，完全なクリッピング処理まで手が回らなかったため，作画範囲を超える線は省略されますので，その点はご容赦ください。作画範囲内での描画は何の問題もありませんので，必要以上に拡大しない使い方をおすすめします。

また，マシン語で実数演算のルーチンを組むことはとても大変なことなので，F-BASIC のROM内ルーチンを使用しました。F-BASIC (Ver.3) は，幸いにマシン語エントリーのアドレスまで同一ですので各機種で走らせることができます。

TILT やSPIN，または図の大きさ等を操作するキーはVer 2 と同じですので前節を参照してください。ただし，作画範囲はVer 2 と異なり，横640ドットをフルに使用しています。そ

のため，角度や大きさを表す数値は表示していません。



(図5-27　ハイスピードバージョンにおけるアクションモードの画面)

■ オートアニメーションモード

アクションモード時にAキーを押すとこのモードに入ります。このモードでは立体図をコマ送り風に動かして見ることが可能です。動きを決めるキーは，アクションモードの角度や大きさを設定するキーと同じで，キーが押されると次のキー入力があるまで，回転または移動・拡大などの動作を続けます。回転角度等の単位は PF キーで調整できます。

コマ送りは，パレットと SCREEN を併用したため緑の単色画面になりますが，画面の切り替えはとても自然で，ちらつきなどは起こりません。瞬時に次の画面に変わるため，いかにも動いたという感じが味わえます。

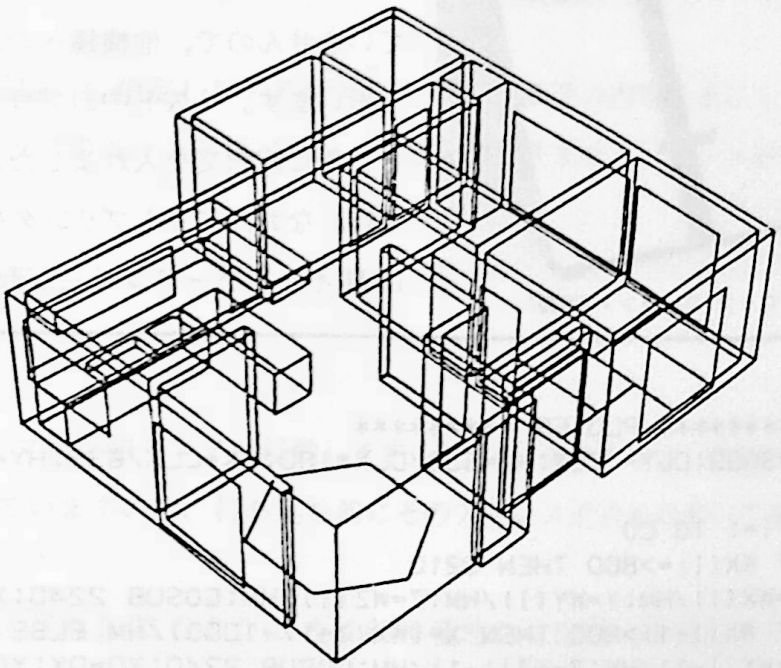
回転方向や移動等は，キー操作によっていつでも変更可能です。←バックスペースキーを押すと，通常のアクションモードに復帰できます。

アニメーション効果はステップ数の少ない立体図ほど有効で，ステップ数の多い複雑な図形になるにつれコマ送りの間隔が長くなります。100 ステップくらいまでが，一応の目安かもしれません。

■ プロッタモード

立体図をプロックに描かせることにより，はじめて実用としての価値が生まれてきます。ドットプリンクでのハードコピーでは，縦と横の比率が変わって印字されるうえ，ドットの分解能も十分ではありません。その点プロックは，正確な描画はむろんですが，分解能に至っては数千ドット×数千ドットという能力を持っています。

最近は，プロックの値段も安くなってきており，1台そろえておくと重宝すると思います。本書で使ったプロックは，ペンの最少可動範囲が0.05ミリと0.1ミリが選べ，約A4サイズまで



(図5-28　プロッタの作図例)

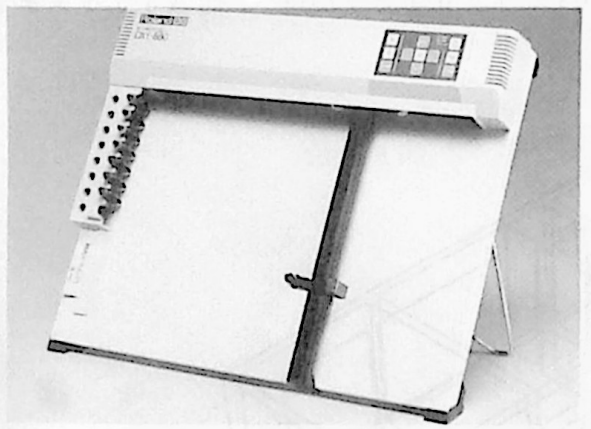
の描画が可能なカシオのFP-1018という機種です。このプロッタは，プリンタの機能も兼ねそなえています。

さて，プロッタへの出力ですが，アクションモード時にPキーを押すと作図を開始します。作図内容はディスプレイの画面と同じものです。テンキーなどで任意の向きの立体図を画面上に高速で描かせ，好みの向きになった時点でPキーを利用してください。なお，プロッタによ

る立体図のクリッピング処理は完全に行なっています。

■　ローランドD.G製プロッタ使用時の変更点

前述のプロッタFP-1018は，たまたま筆者の手元にあったものですが，現在では手に入りにくい機種かもしれません。そこで台数ベースで最も売れているといわれるローランド製のA3判DXYプロッタを使用する場合の，プログラム変更箇所をつぎに示しておきます。リスト5-3の2070～2220行を下記のリストに改めることによりDXYプロッタが使用できます。筆者はDXY880で動作を確認しましたが，DXYシリーズならばどの機種でも動くはずです。



DXY880 プロッタ(ローランドDG製)

プロッタ制御はごく基本的なコマンドしか使っていませんので，他機種への改造も容易にできると思います。リスト中のプロッタ制御文にあたるところに注釈文を入れましたので参考にしてください。なお，ドットプリンタを接続しているときはHキーでハードコピーが取れます。

```
2070 '********** PLOTER *********
2080 CLX=3600:CLY= CLX:HM=639/CLX*100:HX=CLX/639:HY=CLX/175
2090 '
2120 FOR I=1 TO CO
2130    IF WX(I)=>800 THEN 2210
2140    X=WX(I)/HM:Y=WY(I)/HM:Z=WZ(I)/HM:GOSUB 2240:X1=QX:Y1=QY
2150    IF WX(I-1)>800 THEN X=(WX(I-1)-1000)/HM ELSE X=WX(I-1)/HM
2160    Y=WY(I-1)/HM:Z=WZ(I-1)/HM:GOSUB 2240:XO=QX:YO=QY
2170    IF (XO>0)+(XO<CLX)+(YO>0)+(YO<CLY)+(X1>0)+(X1<CLX)+(Y1>0)
        +(Y1<CLY)=-8 THEN 2200
2180    IF (XO<0 AND X1<0) OR (XO>CLX AND X1>CLX) OR (YO<0 AND Y1
        <0) OR (YO>CLY AND Y1>CLY) THEN BEEP B:BEEP O:GOTO 2210
2190    GOSUB 1960:SWAP XO,X1:SWAP YO,Y1:GOSUB 1960
2200    LPRINT"M ";XO;",";CLY-YO;:'------------move
2205    LPRINT"D ";X1;",";CLY-Y1:'-------------draw
2210 NEXT I:LPRINT"H":'----------------------home
2220 GOSUB 2300:CLX=639:CLY=175:RETURN
```

■ マシン語プログラムの入力法

ハイスピードバージョンはベーシックとマシン語からなりたっています。リスト5-3の後半がマシン語プログラムのダンプリストです。

ダンプリストとは，メモリの内容がどのアドレス（メモリの番地）に格納されているかを16進数で書き表したものです。左端の4桁の文字はアドレスを示し，そのアドレスの内容がその右に2桁で表示されています。次のアドレスの内容は，さらにその右に表示されており，1行で8アドレス分（8バイト）のメモリ内容を表わしています。

一番右の2桁の文字はチェックサムで，8バイトを加算して得られた合計の下2桁を表示しており，正しく入力できたかどうかのチェックを行ないます。チェックサムは入力する必要はありません。

マシン語の打ち込みは，指定したアドレスのメモリに16進の内容を正しく入力するわけですが，普通はベーシックのコマンドレベルからでは入力できません。マシン語モニタを起動する必要があります。

● マシン語モニタ

MON ↵

とキーインすると，マシン語モニタが起動します。このプログラムでは，マシン語を&H 6900番地以降に展開させていますので，打ち込む前にそのアドレス近辺をのぞいてみましょう。

D 6900 ↵

とキーインすると，写真5-33のように &H 6900番地から64バイト分のメモリの内容が表示されます。電源 ON 直後は，メモリには何も入っていないため，すべて00の表示になっています。これをリスト5-3のダンプリストの内容に変えていかなければなりません。

メモリの内容を変えるためには，

M 6900 ↵

と打ち込んでください。すると，

6900　00　■

6900：番地　00：現在の内容　■：書き変える内容の入力待ち

の表示になり，データを入力することが可能になります。たとえばここで，12 ↵とキーインすると

6900　00—12

6901　00—■

このように &H 6900 番地には &H 12 が書き込まれるとともに，次の番地 &H 6901 番地の入力待ちとなります。マシン語入力はこの作業を繰り返すわけですが，入力を終了したいときは，16 進以外のキーを打てば入力コマンド M から抜けだせます。

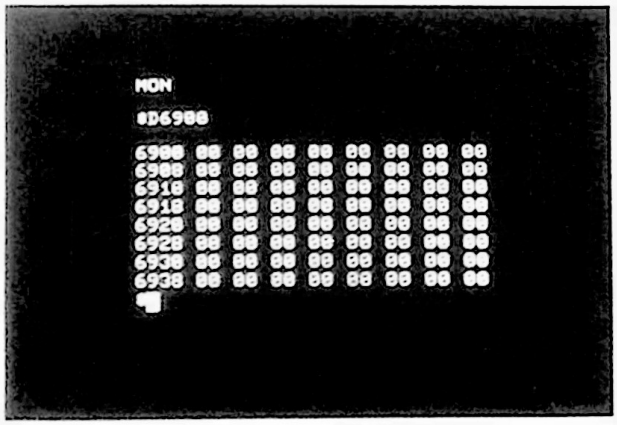
ここで再度，

D 6900 ↵

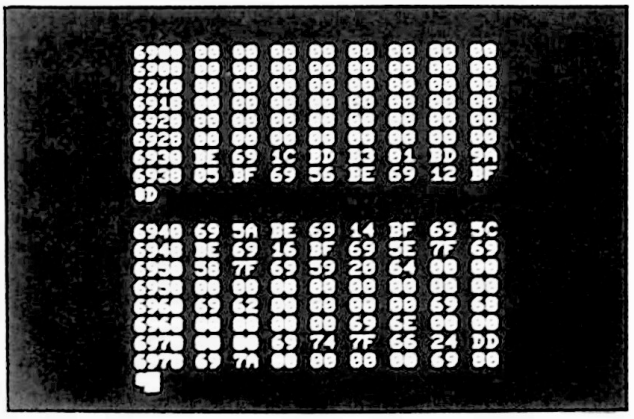
とキーインすると，M コマンド で打ち込んだ内容のメモリダンプが表示され，入力したデータの確認ができます。

以上の入力法は，マシン語モニタの M コマンドを使用する最も手軽な手段ですが，チェックサムの確認まではできません。打ち込みに自信があるのでしたら，この方法でも一向にかまいませんが，市販のマシン語入力ユーティリティを利用するのもひとつの方法です。ユーティリティは，入力しやすいように工夫されていて，16 進の A～F もテンキー部で入力でき，かつ 1 行入力するとチェックサムの表示もしてくれます。

マシン語モニタでの入力は，打ち込んだデータが正しいかどうかをダンプリストと見比べて，ひとつひとつ全部確認しなければならないのですが，チェックサムがあれば，それを見るだけ



(写真 5-33　メモリダンプ)



(写真 5-34　入力後のダンプリスト)

で確認が可能となります。ただし，チェックサムの合致が100％の正しさを証明するとも限らない点は含んでおいてください。

● マシン語の保存

入力を終えたら，

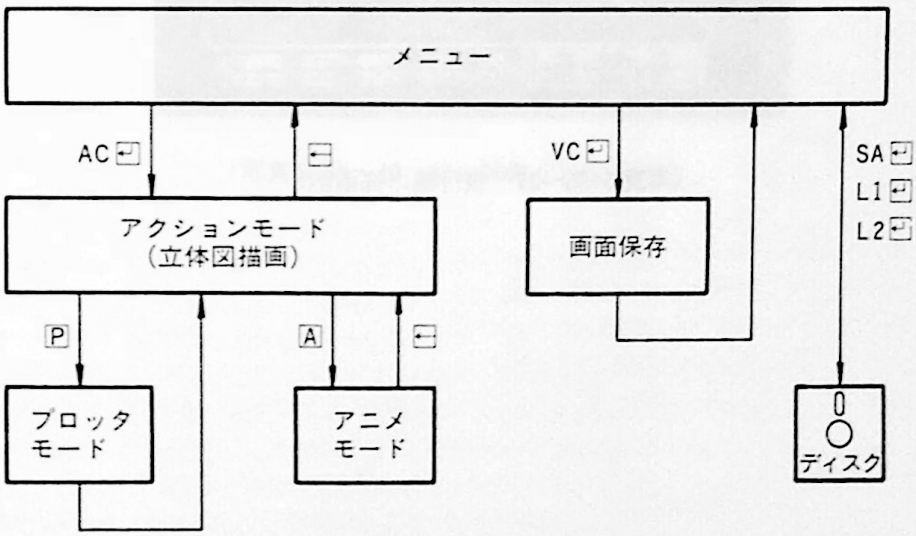
SAVEM　"MSUB 639", &H 6900, &H6 C 76, &H 6900 ↵

として，かならずSAVEしておくとともに，ベーシックプログラムと合わせて，もう1枚コピーを取っておいてください。

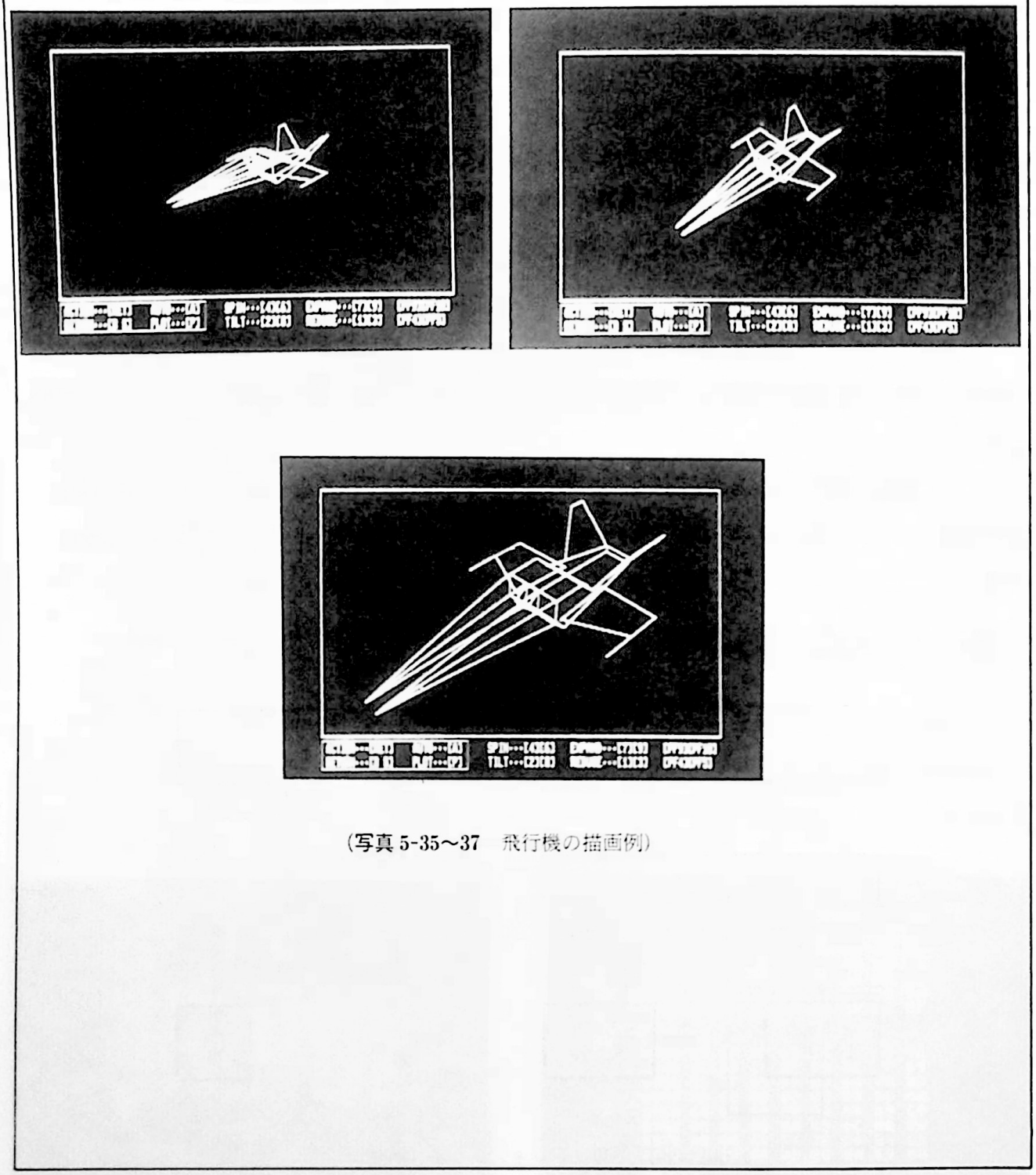
ベーシックプログラムをRUNさせると，このマシン語は自動的にLOADされ，アクションモード時に働き始めますが，1字でも間違えると暴走し，ディスクの破壊にもつながる可能性があります。

マシン語のミスは，エラーメッセージも出さずに，いきなり暴走し始めメモリの内容をメチャメチャにしてしまう仕末に負えないものですから，デバグは細心の注意を払い，充分に行なってください。

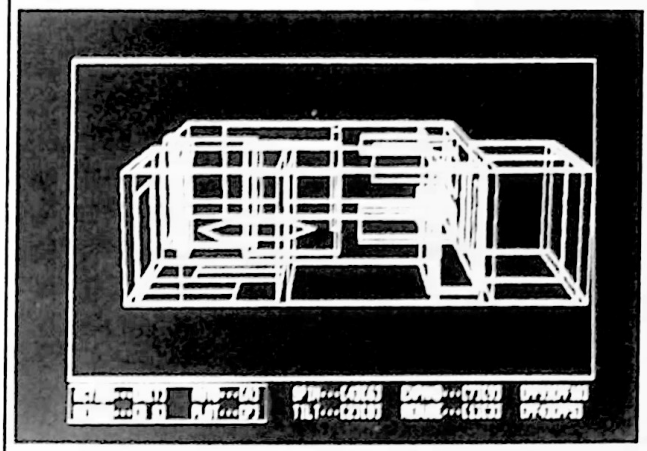
■ ハイスピードバージョンの構成

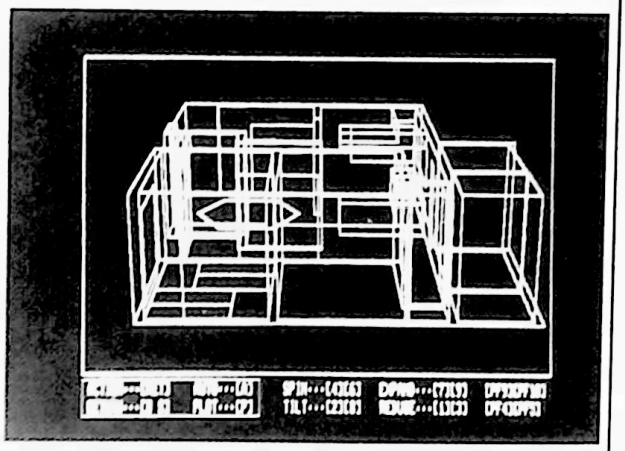


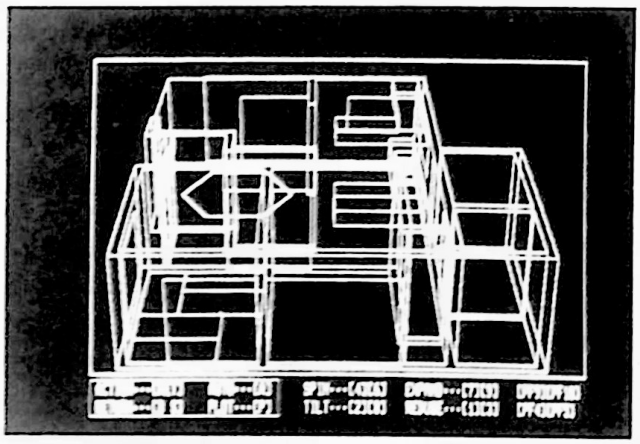
（図5-29）ハイスピードバージョンの構成



(写真 5-35～37　飛行機の描画例)







(写真 5-38～40　家の間取りの描画例)

（リスト5〜3　3D　WORLD　ハイスピードバージョン）

page 1　[File Name : 3DW/HS]

```
1000 '================================
1010 ' F M   3 D   W O R L D  ver2.1
1020 '  HiSpeed Drawing (C)S.Tajima
1030 '================================
1040 CLEAR 500,&H68FF:WIDTH 80,20:CONSOLE 0,18
1050 DIM E%(240):DM$=SPACE$(240)
1060 FOR I=0 TO 17:READ PD:POKE &H6D04+I,PD:NEXT
1070 IF PEEK(&H6930)<>&HBE AND PEEK(&H6938)<>&H05 THEN LOADM"
     MSUB639"
1080 '----------------------------
1090 GOSUB 1370
1100 CLS:CLX=639:CLY=175:KA=8:KZ=.2:CX=322:CY=88
1110 ON KEY(6) GOSUB 2270:ON KEY(9) GOSUB 2370:ON KEY(4) GOSU
     B 2390
1120 ON KEY(10) GOSUB 2310:ON KEY(5) GOSUB 2340:KEY(1)ON
1130 KEY(6)ON:B=1:MC=5:GOTO 1140
1140 '************ MAIN MENU **************
1150 GOSUB 1380:LINE(0,178)-(639,178),PSET,1
1160 ON ERROR GOTO 1310
1170 COLOR 7:LOCATE 0,18:PRINT"3D ACTION····· AC";
1180 LOCATE 25,18:PRINT"SAVE 3D DATA··· SA";
1190 LOCATE 0,19:PRINT"LOAD 1 (NEW)··· L1";
1200 LOCATE 25,19:PRINT"LOAD 2 (CONT)·· L2";
1210 LOCATE 50,18:PRINT"V-RAM COPY····· VC";
1215 LOCATE 50,19:PRINT"END···E";
1217 LOCATE 62,19:PRINT"[ STEP:";CO;"]";
1220 GOSUB 2300
1230 LOCATE 75,19:INPUT D$
1240 IF D$="AC" THEN GOSUB 1390
1250 IF D$="SA" THEN GOSUB 2450
1260 IF D$="L1" THEN GOSUB 2550
1270 IF D$="L2" THEN GOSUB 2570
1280 IF D$="VC" THEN GOSUB 2690
1285 IF D$="E" THEN FOR KY=1 TO 10:KEY(KY) OFF:NEXT:WIDTH80:E
     ND
1290 GOTO 1140
1300 '------------ERROR
1310 CLS:BEEP:IF ERR=9 THEN PRINT"ｺﾚｲｼﾞｮｳ ﾃﾞｰﾀﾊ ﾊｲﾘﾏｾﾝ";:CO=C
     O-1:GOTO 1360
```

page 2　[File Name : 3DW/HS]

```
1320 IF ERR=72 THEN PRINT"ﾃﾞｨｽｸ ｶﾞ OFF ﾃﾞｽ";:GOTO 1360
1330 IF ERR=63 THEN PRINT"ﾌｧｲﾙｶﾞ ﾐﾂｶﾘﾏｾﾝ";:GOTO 1360
1340 IF ERR=64 THEN PRINT"ﾌｧｲﾙﾉ ﾆｼﾞｭｳﾄｳﾛｸ ﾃﾞｽ";:GOTO 1360
1350 IF ERR=80 THEN PRINT"ﾃﾞｰﾀ ｦ LOAD ｼﾃ ｸﾀﾞｻｲ";:GOTO 1360
1360 PRINT"  ﾒﾆｭｰ ﾆ ﾓﾄﾞﾘﾏｽ.    ErrorNo=";ERR:CLOSE:RESUME 1140
1370 DIM WX(1200),WY(1200),WZ(1200):RETURN
1380 FOR K=18 TO 19:LOCATE 0,K:PRINT SPACE$(79);:NEXT:RETURN
1390 '********* ACTION *********
1400 IF CO<2 THEN ERROR 80
1410 KEY(10) ON:KEY(5) ON:KEY(9)ON:KEY(4)ON
1420 GOSUB 2300:LINE(0,178)-(639,178),PSET,0
1430 LOCATE 0,18:PRINT " ACTION･･･[RET]    AUTO･･･[A]      SPI
     N･･･[4][6]   EXPAND･･･[7][9]   [PF9][PF10]"
1440 PRINT              " RETURN･･･[B S]    PLOT･･･[P]      TIL
     T･･･[2][8]   REDUSE･･･[1][3]   [PF4][PF5]";:LINE(0,178)-(
     240,198),PSET,4,B
1450 CX=320:CY=88:OM=10:D=1500:SC=-1.5708:SD=1.5708:S1=SIN(SC
     ):C1=COS(SC)
1460 S2=SIN(SD):C2=COS(SD)
1470 GOSUB 1780
1480 IK$=INKEY$:IFIK$=""THEN 1480
1490 IF IK$="A" THEN GOSUB 1670
1495 IF IK$="H" THEN HARDC1
1500 IF IK$=CHR$(&H08) THEN KEY(10)OFF:KEY(5)OFF:KEY(9)OFF:KE
     Y(4)OFF:GOTO 1140
1510 IF IK$="P" THEN GOSUB 2070
1520 IF IK$=CHR$(13) THEN GOSUB 1790:GOTO 1480
1530 A$=IK$:GOSUB 1540:GOTO 1480
1540 IF A$="3" THEN OM=OM+OM*KZ
1550 IF A$="9" THEN OM=OM-OM*KZ
1560 IF A$="7" THEN D=D+D*KZ
1570 IF A$="1" THEN D=D-D*KZ
1580 IF A$="6" THEN SC=SC+3.14159/KA:S1=SIN(SC):C1=COS(SC)
1590 IF A$="4" THEN SC=SC-3.14159/KA:S1=SIN(SC):C1=COS(SC)
1600 IF A$="8" THEN SD=SD+3.14159/KA:S2=SIN(SD):C2=COS(SD)
1610 IF A$="2" THEN SD=SD-3.14159/KA:S2=SIN(SD):C2=COS(SD)
1620 IF A$=CHR$(&H1C) THEN CX=CX+40
1630 IF A$=CHR$(&H1D) THEN CX=CX-40
1640 IF A$=CHR$(&H1E) THEN CY=CY-20
```

```
page 3    [ File Name : 3DW/HS ]
1650 IF A$=CHR$(&H1F) THEN CY=CY+20
1660 RETURN
1670 POKE &H6C6A,1:GOSUB 1800:SCREEN 1,1:COLOR=(1,4):SCREEN 2
     ,1
1680 LINE(1,1)-(638,174),PSET,0,BF
1690 GOSUB 1740:IF I$=CHR$(&H08) THEN 1770
1700 POKE &H6C6A,2:GOSUB 1800:SCREEN 2,2:COLOR=(2,4)
1710 SCREEN 1,2:LINE(1,1)-(638,174),PSET,0,BF
1720 GOSUB 1740:IF I$=CHR$(&H08) THEN 1770
1730 GOTO 1670
1740 I$=INKEY$:IF I$=""THEN 1760
1750 A$=I$:IF I$=CHR$(&H08) THEN RETURN
1760 GOSUB 1540:RETURN
1770 FOR I=0TO 7:COLOR=(I,I):NEXT:SCREEN7,7:POKE &H6C6A,3:GOS
     UB 1780:RETURN
1780 '********** view work *********
1790 LINE(1,1)-(638,174),PSET,0,BF:LINE(0,0)-(639,175),PSET,7
     ,B
1800 POKE &H6912,VARPTR(WX(1))¥256:POKE &H6913,VARPTR(WX(1))
     MOD 256
1810 POKE &H6914,VARPTR(WY(1))¥256:POKE &H6915,VARPTR(WY(1))
     MOD 256
1820 POKE &H6916,VARPTR(WZ(1))¥256:POKE &H6917,VARPTR(WZ(1))
     MOD 256
1830 POKE &H6918,VARPTR(CX)   ¥256:POKE &H6919,VARPTR(CX)
     MOD 256
1840 POKE &H691A,VARPTR(CY)   ¥256:POKE &H691B,VARPTR(CY)
     MOD 256
1850 POKE &H691C,VARPTR(CO)   ¥256:POKE &H691D,VARPTR(CO)
     MOD 256
1860 POKE &H691E,VARPTR(OM)   ¥256:POKE &H691F,VARPTR(OM)
     MOD 256
1870 POKE &H6920,VARPTR(D)    ¥256:POKE &H6921,VARPTR(D)
     MOD 256
1880 POKE &H6922,VARPTR(S1)   ¥256:POKE &H6923,VARPTR(S1)
     MOD 256
1890 POKE &H6924,VARPTR(S2)   ¥256:POKE &H6925,VARPTR(S2)
     MOD 256
1900 POKE &H6926,VARPTR(C1)   ¥256:POKE &H6927,VARPTR(C1)
```

page 4 [File Name : 3DW/HS]

```
    MOD 256
1910 POKE &H6928,VARPTR(C2)    ¥256:POKE &H6929,VARPTR(C2)
    MOD 256
1920 EXEC &H6930
1930 GOSUB 2300:RETURN
1940 '
1950  '*********** CLIPPING **********
1960 IF X1<0 THEN XX=0:GOTO 1990
1970 IF X1>CLX THEN XX=CLX:GOTO 1990
1980 XX=X1
1990 IF X1=X0 THEN YY=Y1:GOTO2030
2000 YY=Y1-(Y1-Y0)*(X1-XX)/(X1-X0)
2010 IF YY<0 THEN YY=0:GOTO 2040
2020 IF YY>CLY THEN YY=CLY:GOTO2040
2030 X1=XX:Y1=YY:GOTO 2060
2040 IF Y1=Y0 THEN XX=X1:GOTO 2030
2050 XX=X1-(X1-X0)*(Y1-YY)/(Y1-Y0):GOTO 2030
2060 RETURN
2070 '********** PLOTER *********
2080 CLX=1920:CLY=CLX:HM=639/CLX*100:HX=CLX/639:HY=CLX/175
2090 LPRINT CHR$(&H1C);CHR$(&H25):'----------graphic mode
2100 LPRINT"O";0;",";CLY:'--------------------set orign
2110 LPRINT"A";0;",";0;",";CLX;",";-CLY:'-----quad(box)
2120 FOR I=1 TO CO
2130 IF WX(I)=>800 THEN 2210
2140 X=WX(I)/HM:Y=WY(I)/HM:Z=WZ(I)/HM:GOSUB 2240:X1=QX:Y1=QY
2150 IF WX(I-1)>800 THEN X=(WX(I-1)-1000)/HM ELSE X=WX(I-1)/H
    M
2160 Y=WY(I-1)/HM:Z=WZ(I-1)/HM:GOSUB 2240:X0=QX:Y0=QY
2170 IF (X0>0)+(X0<CLX)+(Y0>0)+(Y0<CLY)+(X1>0)+(X1<CLX)+(Y1>0
    )+(Y1<CLY)=-8 THEN2200
2180 IF (X0<0 AND X1<0) OR (X0>CLX AND X1>CLX) OR (Y0<0 AND Y
    1<0) OR (Y0>CLY AND Y1>CLY) THEN BEEP B:BEEP 0:GOTO 2210
2190 GOSUB 1960:SWAP X0,X1:SWAP Y0,Y1:GOSUB 1960
2200 LPRINT"D ";X0;",";-Y0;",";X1;",";-Y1:'---draw
2210 NEXT I:LPRINT"M"0;",";-CLY:'-------------move
2220 GOSUB 2300:CLX=639:CLY=175:RETURN
2230 '-----------------------------------
2240 ZC=OM*HX-X*C1*S2-Y*S1*S2-Z*C2:QX=(D*HX*(-X*S1+Y*C1)/ZC)+
```

```
page 5    [ File Name : 3DW/HS ]
    CX*HX:QY=CY*HY-(D*HX*(-X*C1*C2-Y*S1*C2+Z*S2)/ZC):RETURN
2250 '
2260 '******** BEEP SWICH ********
2270 B=B+1:IF B=2 THEN B=0
2280 RETURN
2290 '--------------------
2300 BEEP B:FOR BT=0 TO 50:NEXT:BEEP 0:RETURN
2310 '****** ANGLE. SCALE. ********
2320 KA=KA/2:IF KA<.1 THEN KA=1
2330 GOSUB 2360:RETURN
2340 KA=KA*2:IF KA>128 THEN KA=128
2350 GOSUB 2360:RETURN
2360 AN=3.14159/KA/(3.14159/180):RETURN
2370 KZ=KZ*2:IF KZ>1 THEN KZ=.8
2380 GOSUB 2410:RETURN
2390 KZ=KZ/2:IF KZ<.1 THEN KZ=.1
2400 GOSUB 2410:RETURN
2410 SL=KZ*100:RETURN
2420 '-----------------
2430 LOCATE 58,17:PRINT"BUSY !!";:RETURN
2440 '
2450 '********** 3D DATA SAVE ************
2460 GOSUB 1380:COLOR 5:LOCATE 0,18:INPUT"FILE NAME";TITLE$:I
    F LEN(TITLE$)>8 THEN BEEP:GOTO 2460
2470 TITLE$=TITLE$+SPACE$(8-LEN(TITLE$)):LOCATE 25,18:PRINT"R
    EADY D E S K !"
2480 LOCATE 50,18:PRINT"HIT RETURN KEY":COLOR 2
2490 BEEP 0:C$=INKEY$:LOCATE 0,19:PRINT"[ DISK <--- DATA ]";
2500 FOR BB=0 TO 50:NEXT:BEEP 1:IF C$ <> CHR$(13) THEN LOCATE
     0,19:PRINT"                    ";:GOTO 2490
2510 LOCATE 50,18:PRINT"                    ";:BEEP 0:OPEN"O",#1,TI
    TLE$
2520 PRINT#1,TITLE$:PRINT#1,CO:FOR S=1 TO CO:PRINT#1,WX(S)
2530 PRINT#1,WY(S):PRINT#1,WZ(S):LOCATE 50,19:PRINT S;:NEXT S
2540 CLOSE #1:RETURN
2550 '*********** LOAD 1 ***********
2560 CO=0:SI=CO:GOSUB 2600:RETURN
2570 '*********** LOAD 2 ***********
2580 SI=CO:GOSUB 2600:RETURN
```

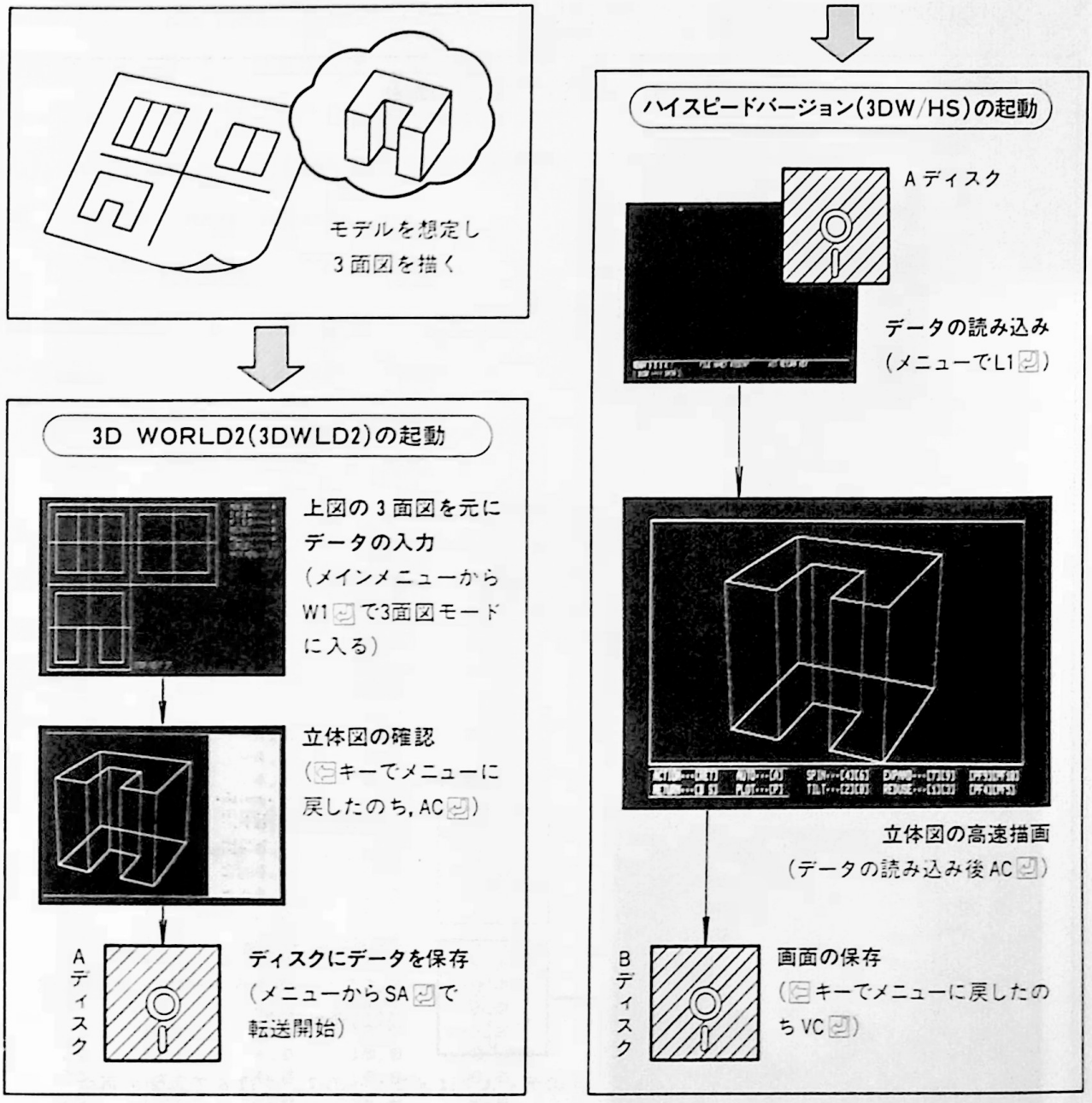
```
page 6    [ File Name : 3DW/HS ]
-------------------------------------------------------------------------------
2590 '------------------------------
2600 GOSUB 1380:COLOR 5:LOCATE 0,18:PRINT"READY D E S K !":LO
     CATE 25,18:INPUT"FILE NAME";TITLE$:IF LEN(TITLE$)>8 THEN
     BEEP:GOTO 2600
2610 TITLE$=TITLE$+SPACE$(8-LEN(TITLE$)):LOCATE 50,18:PRINT"H
     IT RETURN KEY";
2620 BEEP 0:C$=INKEY$:LOCATE 0,19:PRINT"[ DISK ---> DATA ]";
2630 FOR BB=0 TO 50:NEXT:BEEP 1:IF C$ <> CHR$(13) THEN LOCATE
      0,19:PRINT"                          ";:GOTO 2620
2640 LOCATE 50,18:PRINT"                    ";:BEEP 0:OPEN"I",#1,TI
     TLE$
2650 INPUT#1,TITLE$:LOCATE 25,19:PRINT "Found:";TITLE$;:INPUT
     #1,CT:CO=CO+CT:FOR L=SI+1 TO CO:INPUT#1,WX(L)
2660 INPUT#1,WY(L):INPUT#1,WZ(L):LOCATE 50,19:PRINT L;:NEXT L
2670 '
2680 CLOSE #1:RETURN
2690 '***************** V-RAM COPY SUB *****************
2700 GOSUB 1380:LOCATE 0,18
2710 INPUT"SAVE/FILE NAME ";FIN$
2720 OPEN"O",#1,FIN$
2730 FOR Y=0 TO 175:LOCATE 0,19:PRINT"lineNo=";Y;
2740 GET@A(0,Y)-(639,Y),E%,G
2750 POKE &H6D00,VARPTR(E%(0)) ¥ 256
2760 POKE &H6D01,VARPTR(E%(0)) MOD 256
2770 POKE &H6D02,PEEK(VARPTR(DM$)+1)
2780 POKE &H6D03,PEEK(VARPTR(DM$)+2)
2790 EXEC&H6D04:PRINT#1,DM$;:NEXT Y:CLOSE#1:RETURN
2800 '================ MACHINE DATA ===============
2810 DATA 198,240,190,109,0,16,190,109,2,166
2820 DATA 128,167,160,192,1,38,248,57
```

3D WORLD ハイスピードバージョンのマシン語プログラム

```
add   +0 +1 +2 +3 +4 +5 +6 +7 sum
6900  00 00 00 00 00 00 00 00 :00
6908  00 00 00 00 00 00 00 00 :00
6910  00 00 00 00 00 00 00 00 :00
6918  00 00 00 00 00 00 00 00 :00
6920  00 00 00 00 00 00 00 00 :00
6928  00 00 00 00 00 00 00 00 :00
6930  BE 69 1C BD B3 01 BD 9A :0B
6938  05 BF 69 56 BE 69 12 BF :7B

6940  69 5A BE 69 14 BF 69 5C :82
6948  BE 69 16 BF 69 5E 7F 69 :AB
6950  58 7F 69 59 20 64 00 00 :1D
6958  00 00 00 00 00 00 00 00 :00
6960  69 62 00 00 00 00 69 68 :9C
6968  00 00 00 00 69 6E 00 00 :D7
6970  00 00 69 74 7F 66 24 DD :C3
6978  69 7A 00 00 00 00 69 80 :CC

6980  00 00 00 00 69 86 00 00 :EF
6988  00 00 69 8C 00 00 00 00 :F5
6990  69 92 00 00 00 00 69 98 :FC
6998  00 00 00 00 69 9E 00 00 :07
69A0  00 00 69 A4 00 00 00 00 :0D
69A8  69 AA 00 00 00 00 69 B0 :2C
69B0  8A 7A 00 00 69 B6 87 48 :F2
69B8  00 00 FC 69 58 C3 00 01 :81

69C0  FD 69 58 10 B3 69 56 10 :50
69C8  22 00 A6 BE 69 5A BD B3 :B9
69D0  01 BD 9A 05 8C 03 20 10 :1C
69D8  2E 00 78 BE 69 5A BD B3 :97
69E0  01 17 00 B2 BE 69 66 BD :14
69E8  B3 01 BE 69 84 BD B3 37 :06
69F0  BE 69 6C BD B3 01 BE 69 :2B
69F8  8A BD B3 37 FC 69 5A 83 :73

6A00  00 04 FD 69 5A FC 69 5C :85
6A08  83 00 04 FD 69 5C FC 69 :AE
6A10  5E 83 00 04 FD 69 5E 17 :C0
6A18  00 58 BE 69 66 BD B3 01 :56
6A20  BE 69 78 BD B3 37 BE 69 :6D
6A28  6C BD B3 01 BE 69 7E BD :3F
6A30  B3 37 FC 69 5A C3 00 08 :74
6A38  FD 69 5A FC 69 5C C3 00 :44

6A40  08 FD 69 5C FC 69 5E C3 :50
6A48  00 08 FD 69 5E 17 01 A3 :87
6A50  16 FF 67 FC 69 5A C3 00 :FE
6A58  04 FD 69 5A FC 69 5C C3 :48
6A60  00 04 FD 69 5C FC 69 5E :89
6A68  C3 00 04 FD 69 5E 16 FF :A0
6A70  49 39 BE 69 5A BD B3 01 :74
6A78  BD 9A 05 8C 03 20 2E 08 :41

6A80  BE 69 5A BD B3 01 20 0E :20
6A88  BE 69 AE BD B3 01 BE 69 :6D
6A90  5A BD AF 11 20 00 BE 69 :1E
6A98  9C BD B3 37 BE 69 B4 BD :DB
6AA0  B3 01 BE 69 9C BD B2 34 :1A
6AA8  BE 69 9C BD B3 37 BE 69 :91
6AB0  B4 BD B3 01 BE 69 5C BD :65
6AB8  B2 34 BE 69 A2 BD B3 37 :56

6AC0  BE 69 B4 BD B3 01 BE 69 :73
6AC8  5E BD B2 34 BE 69 A8 BD :8D
6AD0  B3 37 BE 69 9C BD B3 01 :1E
6AD8  BD BD C9 BE 69 26 BD B0 :FD
6AE0  EE BE 69 24 BD B0 EE BE :52
6AE8  69 90 BD B3 37 BE 69 A2 :69
6AF0  BD B3 01 BD BD C9 BE 69 :DB
6AF8  22 BD B0 EE BE 69 24 BD :85

6B00  B0 EE BE 69 96 BD B3 37 :02
6B08  BE 69 A8 BD B3 01 BD BD :BA
6B10  C9 BE 69 28 BD B0 EE BE :31
6B18  69 96 BD AF 1A BE 69 90 :3C
6B20  BD AF 1A BE 69 1E BD AF :37
6B28  1A BE 69 60 BD B3 37 BE :06
6B30  69 9C BD B3 01 BD BD C9 :B9
6B38  BE 69 22 BD B0 EE BE 69 :CB

6B40  90 BD B3 37 BE 69 A2 BD :BD
6B48  B3 01 BE 69 26 BD B0 EE :5C
6B50  BE 69 90 BD AF 1A BE 69 :64
6B58  20 BD B0 EE BE 69 90 BD :EF
6B60  B3 37 BE 69 60 BD B3 01 :E2
6B68  BE 69 90 BD B2 34 BE 69 :81
6B70  18 BD AF 1A BE 69 66 BD :E8
6B78  B3 37 BE 69 9C BD B3 01 :1E

6B80  BD BD C9 BE 69 26 BD B0 :FD
6B88  EE BE 69 28 BD B0 EE BE :56
6B90  69 90 BD B3 37 BE 69 A2 :69
6B98  BD B3 01 BD BD C9 BE 69 :DB
6BA0  22 BD B0 EE BE 69 28 BD :89
6BA8  B0 EE BE 69 96 BD B3 37 :02
6BB0  BE 69 A8 BD B3 01 BE 69 :67
6BB8  24 BD B0 EE BE 69 96 BD :F9

6BC0  AF 1A BE 69 90 BD AF 1A :06
6BC8  BE 69 20 BD B0 EE BE 69 :C9
6BD0  90 BD B3 37 BE 69 60 BD :7B
6BD8  B3 01 BE 69 90 BD B2 34 :0E
6BE0  BE 69 72 BD B0 EE BE 69 :1B
6BE8  1A BD AF 11 BE 69 6C BD :E7
6BF0  B3 37 39 BE 69 78 BD B3 :32
6BF8  01 BD 9A 05 BF 6C 6D BE :B3

6C00  69 7E BD B3 01 BD 9A 05 :B4
6C08  BF 6C 6F BE 69 84 BD B3 :B5
6C10  01 BD 9A 05 BF 6C 71 BE :B7
6C18  69 8A BD B3 01 BD 9A 05 :C0
6C20  BF 6C 73 BE 6C 6D 8C 00 :C1
6C28  00 2D 4A 8C 02 7F 2E 45 :F7
6C30  BE 6C 6F 8C 00 00 2D 3D :8F
6C38  8C 00 AF 2E 38 BE 6C 71 :3C

6C40  8C 00 00 2D 30 8C 02 7F :F6
6C48  2E 2B BE 6C 73 8C 00 00 :82
6C50  2D 23 8C 00 AF 2E 1E 8E :65
6C58  6C 5F BD 00 DE 20 16 10 :AC
6C60  00 6C 67 00 0E 00 00 00 :E1
6C68  00 16 03 00 02 00 00 00 :1B
6C70  00 00 00 00 00 39 00 00 :39
6C78  00 00 00 00 00 00 00 00 :00
```

立体図の陰線消去法とペイント法

3D WORLD2とそのハイスピードバージョンおよび第3章のスケッチボード2プログラムは，保存したデータのやりとりが可能です。この3つのプログラムを利用することにより，陰影をつけた立体図の作成ができます。

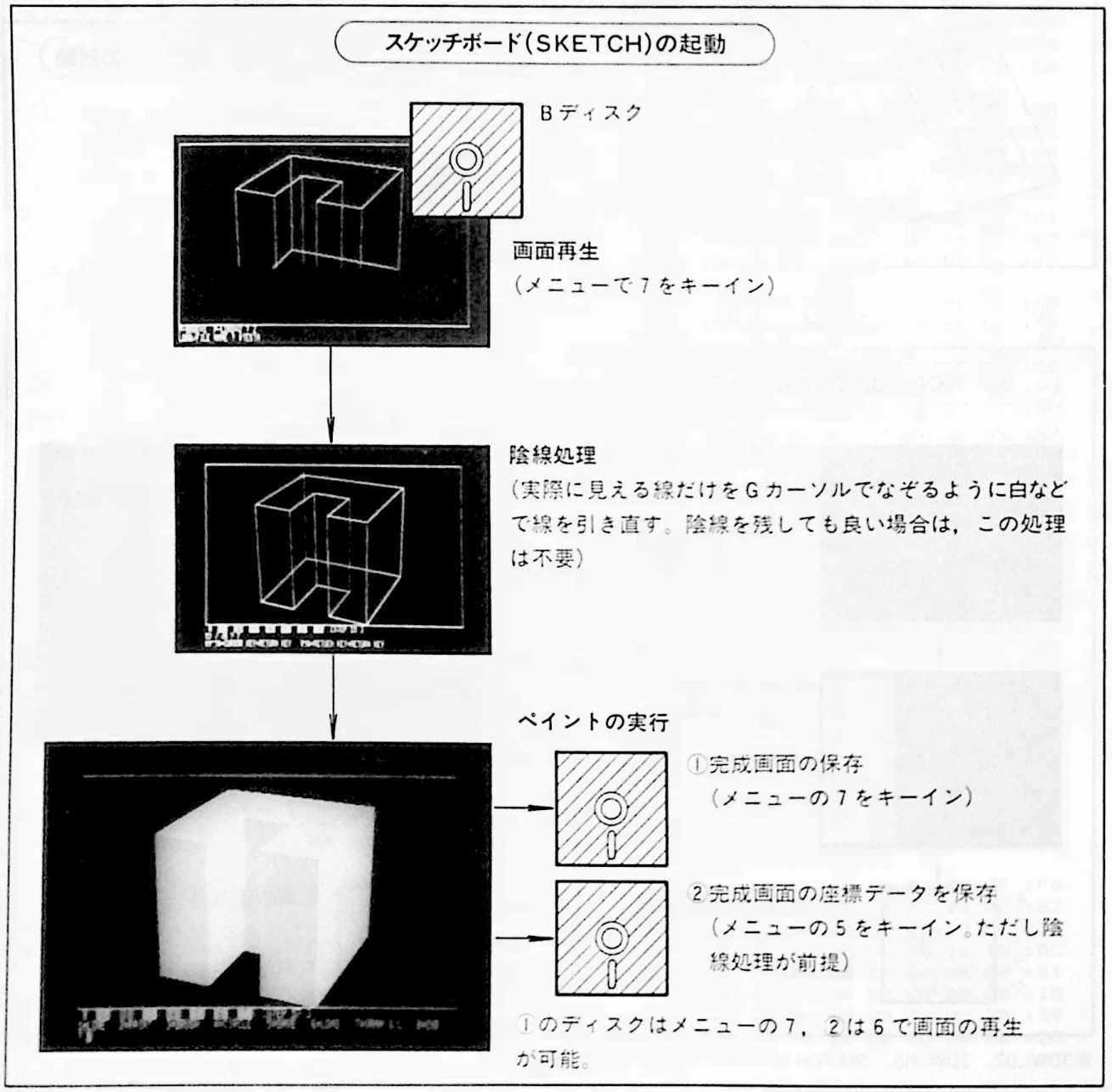


※3DWLD2，3DW/HS，SKETCHはプログラムのファイル名

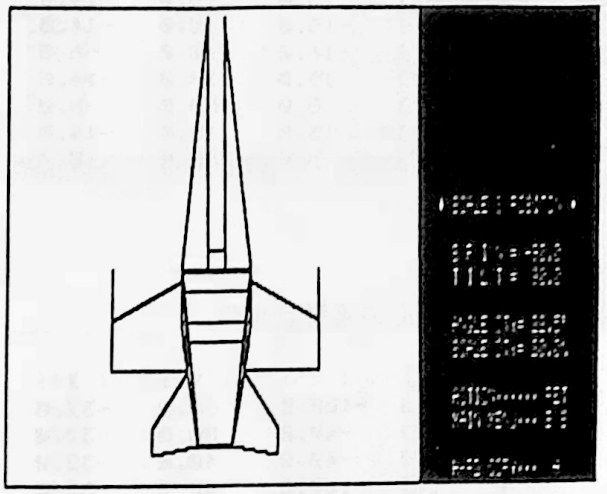
次にその手順を紹介します。

この方法で作成した立体図をいくつかディスクに保存しておくと，わかりやすい立体図の提供がす早く行なえ，建築やデザインのプレゼンテーションには最適です。

またさまざまな大きさや角度からの立体図を作成し，個人のライブラリーとして保存するのも楽しいことかもしれません。手間をかければその分だけ良い画質が得られます。ワイヤーフレームの立体図にあきたときは，この方法を試してください。



3次元データサンプル



3D WORLD Ver2で使用したモデルの3次元データを以下に収載しました。このリストはメインメニューのLD（LLIST DATA）で打ち出したもので，左端からステップNo（これは打ち込む必要はありません），[*] PENSET，[/] DRAWの選択，X座標，Y座標，Z座標の順で並んでいます。

プログラムを起動し，メインメニューでW 1 ↵としたのち，リストに従って[*]か[/]キーを押したうえ，X, Y, Zの順で座標値を入力してください。

アニメ風ジェット機のサンプルデータ

（リスト5-4　写真5-35で示した飛行機の3次元座標リスト）

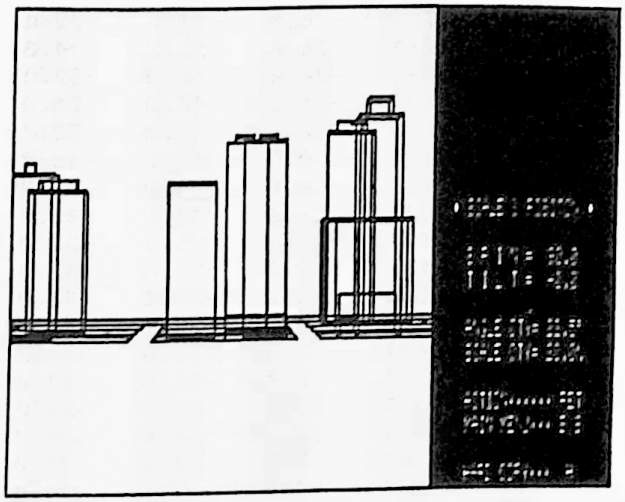
No.		(X)	(Y)	(Z)
1	[*]	-16.0	10.0	8.0
2	[/]	-16.0	30.0	8.0
3	[*]	16.0	10.0	8.0
4	[/]	16.0	30.0	8.0
5	[*]	16.0	10.0	8.0
6	[/]	-16.0	10.0	8.0
7	[*]	16.0	30.0	8.0
8	[/]	-16.0	30.0	8.0
9	[*]	-4.0	10.0	-6.0
10	[/]	-4.0	2.0	2.0
11	[/]	-4.0	10.0	8.0
12	[*]	4.0	10.0	-6.0
13	[/]	4.0	2.0	2.0
14	[/]	4.0	10.0	8.0
15	[*]	4.0	10.0	-6.0
16	[/]	-4.0	10.0	-6.0
17	[*]	4.0	2.0	2.0
18	[/]	-4.0	2.0	2.0
19	[*]	4.0	10.0	8.0
20	[/]	-4.0	10.0	8.0
21	[*]	4.0	10.0	8.0
22	[/]	4.0	-90.0	0.0
23	[/]	4.0	10.0	-6.0
24	[/]	4.0	10.0	8.0
25	[*]	-4.0	10.0	8.0
26	[/]	-4.0	-90.0	0.0
27	[/]	-4.0	10.0	-6.0
28	[/]	-4.0	10.0	8.0
29	[*]	-16.0	10.0	8.0
30	[/]	-4.0	-90.0	0.0
31	[/]	-16.0	10.0	-6.0
32	[/]	-16.0	10.0	8.0
33	[/]	16.0	10.0	8.0
34	[/]	16.0	10.0	-6.0
35	[/]	-16.0	10.0	-6.0
36	[*]	16.0	10.0	8.0
37	[/]	4.0	-90.0	0.0
38	[/]	16.0	10.0	-6.0
39	[/]	16.0	10.0	-6.0
40	[*]	16.0	30.0	8.0
41	[/]	15.0	38.0	0.0
42	[/]	-15.0	38.0	0.0
43	[/]	-16.0	30.0	8.0
44	[*]	-15.0	38.0	0.0
45	[/]	-8.0	80.0	-4.0
46	[/]	-28.0	80.0	20.0
47	[/]	-28.0	70.0	20.0
48	[/]	-13.0	50.0	-1.2
49	[*]	15.0	38.0	0.0
50	[/]	8.0	80.0	-4.0
51	[/]	28.0	80.0	20.0
52	[/]	28.0	70.0	20.0
53	[/]	13.0	50.0	-1.2
54	[*]	-8.0	80.0	-4.0

55	[/]	-8.0	80.0	-8.0
56	[/]	8.0	80.0	-8.0
57	[/]	8.0	80.0	-4.0
58	[/]	-8.0	80.0	-4.0
59	[*]	16.0	10.0	-6.0
60	[/]	15.0	18.0	-14.0
61	[/]	-15.0	18.0	-14.0
62	[/]	-16.0	10.0	-6.0
63	[*]	15.0	18.0	-14.0
64	[/]	8.0	80.0	-8.0
65	[*]	-15.0	18.0	-14.0
66	[/]	-8.0	80.0	-8.0
67	[*]	13.0	50.0	-6.0
68	[/]	48.0	50.0	-6.0
69	[/]	48.0	10.0	-6.0
70	[/]	48.0	30.0	-6.0
71	[/]	15.2	14.0	-6.0
72	[/]	13.0	50.0	-6.0
73	[*]	-13.0	50.0	-6.0
74	[/]	-48.0	50.0	-6.0
75	[/]	-48.0	10.0	-6.0
76	[/]	-48.0	30.0	-6.0
77	[/]	-15.2	14.0	-6.0
78	[/]	-13.0	50.0	-6.0

副都心の高層ビル群

No.		(X)	(Y)	(Z)
1	[*]	-100.0	80.0	-32.0
2	[/]	-40.0	80.0	-32.0
3	[/]	-40.0	40.0	-32.0
4	[/]	-100.0	40.0	-32.0
5	[/]	-100.0	80.0	-32.0
6	[*]	-30.0	80.0	-32.0
7	[/]	30.0	80.0	-32.0
8	[/]	30.0	40.0	-32.0
9	[/]	-30.0	40.0	-32.0
10	[/]	-30.0	80.0	-32.0
11	[*]	40.0	80.0	-32.0
12	[/]	100.0	80.0	-32.0
13	[/]	100.0	40.0	-32.0
14	[/]	40.0	40.0	-32.0
15	[/]	40.0	80.0	-32.0
16	[*]	-100.0	30.0	-32.0
17	[/]	-40.0	30.0	-32.0
18	[/]	-40.0	-10.0	-32.0
19	[/]	-100.0	-10.0	-32.0
20	[/]	-100.0	30.0	-32.0
21	[*]	-30.0	30.0	-32.0
22	[/]	30.0	30.0	-32.0
23	[/]	30.0	-10.0	-32.0
24	[/]	-30.0	-10.0	-32.0
25	[/]	-30.0	30.0	-32.0
26	[*]	40.0	30.0	-32.0
27	[/]	100.0	30.0	-32.0
28	[/]	100.0	-10.0	-32.0
29	[/]	40.0	-10.0	-32.0
30	[/]	40.0	30.0	-32.0
31	[*]	-100.0	-20.0	-32.0
32	[/]	-40.0	-20.0	-32.0
33	[/]	-40.0	-60.0	-32.0
34	[/]	-100.0	-60.0	-32.0
35	[/]	-100.0	-20.0	-32.0
36	[*]	-30.0	-20.0	-32.0
37	[/]	30.0	-20.0	-32.0
38	[/]	30.0	-60.0	-32.0
39	[/]	-30.0	-60.0	-32.0
40	[/]	-30.0	-20.0	-32.0
41	[*]	40.0	-20.0	-32.0
42	[/]	100.0	-20.0	-32.0
43	[/]	100.0	-60.0	-32.0
44	[/]	40.0	-60.0	-32.0
45	[/]	40.0	-20.0	-32.0
46	[*]	-100.0	40.0	-28.0
47	[/]	100.0	40.0	-28.0
48	[*]	-100.0	30.0	-28.0
49	[/]	100.0	30.0	-28.0
50	[*]	-100.0	-10.0	-28.0
51	[/]	100.0	-10.0	-28.0
52	[*]	-100.0	-20.0	-28.0
53	[/]	100.0	-20.0	-28.0
54	[*]	56.0	-28.0	-32.0
55	[/]	74.0	-28.0	-32.0
56	[/]	74.0	-52.0	-32.0
57	[/]	56.0	-52.0	-32.0
58	[/]	56.0	-28.0	-32.0
59	[*]	56.0	-28.0	48.0
60	[/]	74.0	-28.0	48.0
61	[/]	74.0	-52.0	48.0
62	[/]	56.0	-52.0	48.0
63	[/]	56.0	-28.0	48.0
64	[*]	56.0	-28.0	-32.0
65	[/]	56.0	-28.0	48.0
66	[*]	74.0	-28.0	-32.0
67	[/]	74.0	-28.0	48.0
68	[*]	74.0	-52.0	-32.0
69	[/]	74.0	-52.0	48.0
70	[*]	56.0	-52.0	-32.0
71	[/]	56.0	-52.0	48.0
72	[*]	56.0	-28.0	-32.0
73	[/]	56.0	-28.0	48.0
74	[*]	60.0	-32.0	48.0
75	[/]	70.0	-32.0	48.0
76	[/]	70.0	-48.0	48.0
77	[/]	60.0	-48.0	48.0
78	[/]	60.0	-32.0	48.0
79	[*]	60.0	-32.0	54.0
80	[/]	70.0	-32.0	54.0
81	[/]	70.0	-48.0	54.0
82	[/]	60.0	-48.0	54.0

83	[/]	60.0	-32.0	54.0
84	[*]	60.0	-32.0	48.0
85	[/]	60.0	-32.0	54.0
86	[*]	70.0	-32.0	48.0
87	[/]	70.0	-32.0	54.0
88	[*]	70.0	-48.0	48.0
89	[/]	70.0	-48.0	54.0
90	[*]	60.0	-48.0	48.0
91	[/]	60.0	-48.0	54.0
92	[*]	60.0	-32.0	48.0
93	[/]	60.0	-32.0	54.0
94	[*]	0.0	-26.0	-30.0
95	[/]	18.0	-26.0	-30.0
96	[/]	18.0	-28.0	-30.0
97	[/]	26.0	-28.0	-30.0
98	[/]	26.0	-34.0	-30.0
99	[/]	8.0	-34.0	-30.0
100	[/]	8.0	-32.0	-30.0
101	[/]	0.0	-32.0	-30.0
102	[/]	0.0	-26.0	-30.0
103	[*]	0.0	-26.0	40.0
104	[/]	18.0	-26.0	40.0
105	[/]	18.0	-28.0	40.0
106	[/]	26.0	-28.0	40.0
107	[/]	26.0	-34.0	40.0
108	[/]	8.0	-34.0	40.0
109	[/]	8.0	-32.0	40.0
110	[/]	0.0	-32.0	40.0
111	[/]	0.0	-26.0	40.0
112	[*]	0.0	-26.0	-30.0
113	[/]	0.0	-26.0	40.0
114	[*]	18.0	-26.0	-30.0
115	[/]	18.0	-26.0	40.0
116	[*]	18.0	-28.0	-30.0
117	[/]	18.0	-28.0	40.0
118	[*]	26.0	-28.0	-30.0
119	[/]	26.0	-28.0	40.0
120	[*]	26.0	-34.0	-30.0
121	[/]	26.0	-34.0	40.0
122	[*]	8.0	-34.0	-30.0
123	[/]	8.0	-34.0	40.0
124	[*]	8.0	-32.0	-30.0
125	[/]	8.0	-32.0	40.0
126	[*]	0.0	-32.0	-30.0
127	[/]	0.0	-32.0	40.0
128	[*]	0.0	-26.0	-30.0
129	[/]	0.0	-26.0	40.0
130	[*]	4.0	-29.0	40.0
131	[/]	12.0	-29.0	40.0
132	[/]	12.0	-31.0	40.0
133	[/]	4.0	-31.0	40.0
134	[/]	4.0	-29.0	40.0
135	[*]	4.0	-29.0	42.0
136	[/]	12.0	-29.0	42.0
137	[/]	12.0	-31.0	42.0
138	[/]	4.0	-31.0	42.0
139	[/]	4.0	-29.0	42.0
140	[*]	4.0	-29.0	40.0
141	[/]	4.0	-29.0	42.0
142	[*]	12.0	-29.0	40.0
143	[/]	12.0	-29.0	42.0



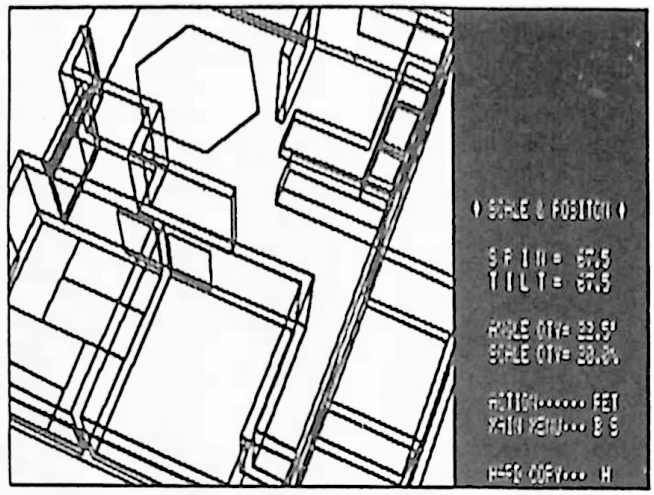
144	[*]	12.0	-31.0	40.0
145	[/]	12.0	-31.0	42.0
146	[*]	4.0	-31.0	40.0
147	[/]	4.0	-31.0	42.0
148	[*]	4.0	-29.0	40.0
149	[/]	4.0	-29.0	42.0
150	[*]	14.0	-29.0	40.0
151	[/]	22.0	-29.0	40.0
152	[/]	22.0	-31.0	40.0
153	[/]	14.0	-31.0	40.0
154	[/]	14.0	-29.0	40.0
155	[*]	14.0	-29.0	42.0
156	[/]	22.0	-29.0	42.0
157	[/]	22.0	-31.0	42.0
158	[/]	14.0	-31.0	42.0
159	[/]	14.0	-29.0	42.0
160	[*]	14.0	-29.0	40.0
161	[/]	14.0	-29.0	42.0
162	[*]	22.0	-29.0	40.0
163	[/]	22.0	-29.0	42.0
164	[*]	22.0	-31.0	40.0
165	[/]	22.0	-31.0	42.0
166	[*]	14.0	-31.0	40.0
167	[/]	14.0	-31.0	42.0
168	[*]	14.0	-29.0	40.0
169	[/]	14.0	-29.0	42.0
170	[*]	-24.0	-46.0	-32.0
171	[/]	-4.0	-46.0	-32.0
172	[/]	-4.0	-52.0	-32.0
173	[/]	-24.0	-52.0	-32.0
174	[/]	-24.0	-46.0	-32.0
175	[*]	-24.0	-46.0	24.0
176	[/]	-4.0	-46.0	24.0
177	[/]	-4.0	-52.0	24.0
178	[/]	-24.0	-52.0	24.0
179	[/]	-24.0	-46.0	24.0
180	[*]	-24.0	-46.0	-32.0
181	[/]	-24.0	-46.0	24.0
182	[*]	-4.0	-46.0	-32.0
183	[/]	-4.0	-46.0	24.0
184	[*]	-4.0	-52.0	-32.0
185	[/]	-4.0	-52.0	24.0

186	[*]	-24.0	-52.0	-32.0
187	[/]	-24.0	-52.0	24.0
188	[*]	-24.0	-46.0	-32.0
189	[/]	-24.0	-46.0	24.0
190	[*]	0.0	-32.0	-32.0
191	[/]	8.0	-32.0	-32.0
192	[/]	8.0	-34.0	-32.0
193	[/]	26.0	-34.0	-32.0
194	[/]	26.0	-56.0	-32.0
195	[/]	0.0	-56.0	-32.0
196	[/]	0.0	-32.0	-32.0
197	[*]	0.0	-32.0	-24.0
198	[/]	8.0	-32.0	-24.0
199	[/]	8.0	-34.0	-24.0
200	[/]	26.0	-34.0	-24.0
201	[/]	26.0	-56.0	-24.0
202	[/]	0.0	-56.0	-24.0
203	[/]	0.0	-32.0	-24.0
204	[*]	0.0	-32.0	-32.0
205	[/]	0.0	-32.0	-24.0
206	[*]	8.0	-32.0	-32.0
207	[/]	8.0	-32.0	-24.0
208	[*]	8.0	-34.0	-32.0
209	[/]	8.0	-34.0	-24.0
210	[*]	26.0	-34.0	-32.0
211	[/]	26.0	-34.0	-24.0
212	[*]	26.0	-56.0	-32.0
213	[/]	26.0	-56.0	-24.0
214	[*]	0.0	-56.0	-32.0
215	[/]	0.0	-56.0	-24.0
216	[*]	0.0	-32.0	-32.0
217	[/]	0.0	-32.0	-24.0
218	[*]	-92.0	-26.0	-32.0
219	[/]	-72.0	-26.0	-32.0
220	[/]	-72.0	-48.0	-32.0
221	[/]	-92.0	-48.0	-32.0
222	[/]	-92.0	-26.0	-32.0
223	[*]	-92.0	-26.0	28.0
224	[/]	-72.0	-26.0	28.0
225	[/]	-72.0	-48.0	28.0
226	[/]	-92.0	-48.0	28.0
227	[/]	-92.0	-26.0	28.0
228	[*]	-92.0	-26.0	-32.0
229	[/]	-92.0	-26.0	28.0
230	[*]	-72.0	-26.0	-32.0
231	[/]	-72.0	-26.0	28.0
232	[*]	-72.0	-48.0	-32.0
233	[/]	-72.0	-48.0	28.0
234	[*]	-92.0	-48.0	-32.0
235	[/]	-92.0	-48.0	28.0
236	[*]	-92.0	-26.0	-32.0
237	[/]	-92.0	-26.0	28.0
238	[*]	-84.0	-36.0	28.0
239	[/]	-80.0	-36.0	28.0
240	[/]	-80.0	-40.0	28.0
241	[/]	-84.0	-40.0	28.0
242	[/]	-84.0	-36.0	28.0
243	[*]	-84.0	-36.0	32.0
244	[/]	-80.0	-36.0	32.0
245	[/]	-80.0	-40.0	32.0
246	[/]	-84.0	-40.0	32.0
247	[/]	-84.0	-36.0	32.0
248	[*]	-84.0	-36.0	28.0
249	[/]	-84.0	-36.0	32.0
250	[*]	-80.0	-36.0	28.0
251	[/]	-80.0	-36.0	32.0
252	[*]	-80.0	-40.0	28.0
253	[/]	-80.0	-40.0	32.0
254	[*]	-84.0	-40.0	28.0
255	[/]	-84.0	-40.0	32.0
256	[*]	-84.0	-36.0	28.0
257	[/]	-84.0	-36.0	32.0
258	[*]	47.0	15.0	-30.0
259	[/]	63.0	24.0	-30.0
260	[/]	68.0	21.0	-30.0
261	[/]	68.0	3.0	-30.0
262	[/]	63.0	0.0	-30.0
263	[/]	47.0	9.0	-30.0
264	[/]	47.0	15.0	-30.0
265	[*]	47.0	15.0	46.0
266	[/]	63.0	24.0	46.0
267	[/]	68.0	21.0	46.0
268	[/]	68.0	3.0	46.0
269	[/]	63.0	0.0	46.0
270	[/]	47.0	9.0	46.0
271	[/]	47.0	15.0	46.0
272	[*]	47.0	15.0	-30.0
273	[/]	47.0	15.0	46.0
274	[*]	63.0	24.0	-30.0
275	[/]	63.0	24.0	46.0
276	[*]	68.0	21.0	-30.0
277	[/]	68.0	21.0	46.0
278	[*]	68.0	3.0	-30.0
279	[/]	68.0	3.0	46.0
280	[*]	63.0	0.0	-30.0
281	[/]	63.0	0.0	46.0
282	[*]	47.0	9.0	-30.0
283	[/]	47.0	9.0	46.0
284	[*]	47.0	15.0	-30.0
285	[/]	47.0	15.0	46.0
286	[*]	51.0	12.0	46.0
287	[/]	64.0	20.0	46.0
288	[/]	64.0	4.0	46.0
289	[/]	51.0	12.0	46.0
290	[*]	51.0	12.0	50.0
291	[/]	64.0	20.0	50.0
292	[/]	64.0	4.0	50.0
293	[/]	51.0	12.0	50.0
294	[*]	51.0	12.0	46.0
295	[/]	51.0	12.0	50.0
296	[*]	64.0	20.0	46.0
297	[/]	64.0	20.0	50.0
298	[*]	64.0	4.0	46.0
299	[/]	64.0	4.0	50.0
300	[*]	51.0	12.0	46.0
301	[/]	51.0	12.0	50.0
302	[*]	48.0	68.0	-32.0
303	[/]	92.0	68.0	-32.0
304	[/]	92.0	48.0	-32.0
305	[/]	86.0	48.0	-32.0
306	[/]	74.0	62.0	-32.0
307	[/]	66.0	62.0	-32.0

308	[/]	54.0	48.0	-32.0
309	[/]	48.0	48.0	-32.0
310	[/]	48.0	68.0	-32.0
311	[*]	48.0	68.0	12.0
312	[/]	92.0	68.0	12.0
313	[/]	92.0	48.0	12.0
314	[/]	86.0	48.0	12.0
315	[/]	74.0	62.0	12.0
316	[/]	66.0	62.0	12.0
317	[/]	54.0	48.0	12.0
318	[/]	48.0	48.0	12.0
319	[/]	48.0	68.0	12.0
320	[*]	48.0	68.0	-32.0
321	[/]	48.0	68.0	12.0
322	[*]	92.0	68.0	-32.0
323	[/]	92.0	68.0	12.0
324	[*]	92.0	48.0	-32.0
325	[/]	92.0	48.0	12.0
326	[*]	86.0	48.0	-32.0
327	[/]	86.0	48.0	12.0
328	[*]	74.0	62.0	-32.0
329	[/]	74.0	62.0	12.0
330	[*]	66.0	62.0	-32.0
331	[/]	66.0	62.0	12.0
332	[*]	54.0	48.0	-32.0
333	[/]	54.0	48.0	12.0
334	[*]	48.0	48.0	-32.0
335	[/]	48.0	48.0	12.0
336	[*]	48.0	68.0	-32.0
337	[/]	48.0	68.0	12.0
338	[*]	56.0	52.0	-32.0
339	[/]	66.0	62.0	-32.0
340	[/]	74.0	62.0	-32.0
341	[/]	84.0	52.0	-32.0
342	[/]	56.0	52.0	-32.0
343	[*]	56.0	52.0	-20.0
344	[/]	66.0	62.0	-20.0
345	[/]	74.0	62.0	-20.0
346	[/]	84.0	52.0	-20.0
347	[/]	56.0	52.0	-20.0
348	[*]	56.0	52.0	-32.0
349	[/]	56.0	52.0	-20.0
350	[*]	66.0	62.0	-32.0
351	[/]	66.0	62.0	-20.0
352	[*]	74.0	62.0	-32.0
353	[/]	74.0	62.0	-20.0
354	[*]	84.0	52.0	-32.0
355	[/]	84.0	52.0	-20.0
356	[*]	56.0	52.0	-32.0
357	[/]	56.0	52.0	-20.0
358	[*]	-88.0	22.0	-32.0
359	[/]	-64.0	22.0	-32.0
360	[/]	-64.0	2.0	-32.0
361	[/]	-88.0	2.0	-32.0
362	[/]	-88.0	22.0	-32.0
363	[*]	-88.0	22.0	24.0
364	[/]	-64.0	22.0	24.0
365	[/]	-64.0	2.0	24.0
366	[/]	-88.0	2.0	24.0
367	[/]	-88.0	22.0	24.0
368	[*]	-88.0	22.0	-32.0
369	[/]	-88.0	22.0	24.0
370	[*]	-64.0	22.0	-32.0
371	[/]	-64.0	22.0	24.0
372	[*]	-64.0	2.0	-32.0
373	[/]	-64.0	2.0	24.0
374	[*]	-88.0	2.0	-32.0
375	[/]	-88.0	2.0	24.0
376	[*]	-88.0	22.0	-32.0
377	[/]	-88.0	22.0	24.0
378	[*]	-84.0	18.0	24.0
379	[/]	-68.0	18.0	24.0
380	[/]	-68.0	6.0	24.0
381	[/]	-84.0	6.0	24.0
382	[/]	-84.0	18.0	24.0
383	[*]	-84.0	18.0	28.0
384	[/]	-68.0	18.0	28.0
385	[/]	-68.0	6.0	28.0
386	[/]	-84.0	6.0	28.0
387	[/]	-84.0	18.0	28.0
388	[*]	-84.0	18.0	24.0
389	[/]	-84.0	18.0	28.0
390	[*]	-68.0	18.0	24.0
391	[/]	-68.0	18.0	28.0
392	[*]	-68.0	6.0	24.0
393	[/]	-68.0	6.0	28.0
394	[*]	-84.0	6.0	24.0
395	[/]	-84.0	6.0	28.0
396	[*]	-84.0	18.0	24.0
397	[/]	-84.0	18.0	28.0
398	[*]	48.0	-28.0	-32.0
399	[/]	56.0	-28.0	-32.0
400	[/]	56.0	-52.0	-32.0
401	[/]	48.0	-52.0	-32.0
402	[/]	48.0	-28.0	-32.0
403	[*]	48.0	-28.0	-26.0
404	[/]	56.0	-28.0	-26.0
405	[/]	56.0	-52.0	-26.0
406	[/]	48.0	-52.0	-26.0
407	[/]	48.0	-28.0	-26.0
408	[*]	48.0	-28.0	-32.0
409	[/]	48.0	-28.0	-26.0
410	[*]	56.0	-28.0	-32.0
411	[/]	56.0	-28.0	-26.0
412	[*]	56.0	-52.0	-32.0
413	[/]	56.0	-52.0	-26.0
414	[*]	48.0	-52.0	-32.0
415	[/]	48.0	-52.0	-26.0
416	[*]	48.0	-28.0	-32.0
417	[/]	48.0	-28.0	-26.0

家の間取り

No.		(X)	(Y)	(Z)
1	[*]	-60.0	102.0	0.0
2	[/]	62.0	102.0	0.0
3	[/]	62.0	2.0	0.0
4	[/]	102.0	2.0	0.0
5	[/]	102.0	-102.0	0.0
6	[/]	-82.0	-102.0	0.0
7	[/]	-82.0	-18.0	0.0
8	[/]	42.0	-18.0	0.0
9	[/]	42.0	-78.0	0.0
10	[/]	58.0	-78.0	0.0
11	[/]	58.0	98.0	0.0
12	[/]	2.0	98.0	0.0
13	[/]	2.0	60.0	0.0
14	[/]	-2.0	60.0	0.0
15	[/]	-2.0	98.0	0.0
16	[/]	-60.0	98.0	0.0
17	[/]	-60.0	102.0	0.0
18	[*]	-60.0	102.0	52.0
19	[/]	62.0	102.0	52.0
20	[/]	62.0	2.0	52.0
21	[/]	102.0	2.0	52.0
22	[/]	102.0	-102.0	52.0
23	[/]	-82.0	-102.0	52.0
24	[/]	-82.0	-18.0	52.0
25	[/]	42.0	-18.0	52.0
26	[/]	42.0	-78.0	52.0
27	[/]	58.0	-78.0	52.0
28	[/]	58.0	98.0	52.0
29	[/]	2.0	98.0	52.0
30	[/]	2.0	60.0	52.0
31	[/]	-2.0	60.0	52.0
32	[/]	-2.0	98.0	52.0
33	[/]	-60.0	98.0	52.0
34	[/]	-60.0	102.0	52.0
35	[*]	-60.0	102.0	0.0
36	[/]	-60.0	102.0	52.0
37	[*]	62.0	102.0	0.0
38	[/]	62.0	102.0	52.0
39	[*]	62.0	2.0	0.0
40	[/]	62.0	2.0	52.0
41	[*]	102.0	2.0	0.0
42	[/]	102.0	2.0	52.0
43	[*]	102.0	-102.0	0.0
44	[/]	102.0	-102.0	52.0
45	[*]	-82.0	-102.0	0.0
46	[/]	-82.0	-102.0	52.0
47	[*]	-82.0	-18.0	0.0
48	[/]	-82.0	-18.0	52.0
49	[*]	42.0	-18.0	0.0
50	[/]	42.0	-18.0	52.0
51	[*]	42.0	-78.0	0.0
52	[/]	42.0	-78.0	52.0
53	[*]	58.0	-78.0	0.0
54	[/]	58.0	-78.0	52.0



55	[*]	58.0	98.0	0.0
56	[/]	58.0	98.0	52.0
57	[*]	2.0	98.0	0.0
58	[/]	2.0	98.0	52.0
59	[*]	2.0	60.0	0.0
60	[/]	2.0	60.0	52.0
61	[*]	-2.0	60.0	0.0
62	[/]	-2.0	60.0	52.0
63	[*]	-2.0	98.0	0.0
64	[/]	-2.0	98.0	52.0
65	[*]	-60.0	98.0	0.0
66	[/]	-60.0	98.0	52.0
67	[*]	-60.0	102.0	0.0
68	[/]	-60.0	102.0	52.0
69	[*]	62.0	-2.0	0.0
70	[/]	98.0	-2.0	0.0
71	[/]	98.0	-58.0	0.0
72	[/]	62.0	-58.0	0.0
73	[/]	62.0	-2.0	0.0
74	[*]	62.0	-2.0	52.0
75	[/]	98.0	-2.0	52.0
76	[/]	98.0	-58.0	52.0
77	[/]	62.0	-58.0	52.0
78	[/]	62.0	-2.0	52.0
79	[*]	62.0	-2.0	0.0
80	[/]	62.0	-2.0	52.0
81	[*]	98.0	-2.0	0.0
82	[/]	98.0	-2.0	52.0
83	[*]	98.0	-58.0	0.0
84	[/]	98.0	-58.0	52.0
85	[*]	62.0	-58.0	0.0
86	[/]	62.0	-58.0	52.0
87	[*]	62.0	-2.0	0.0
88	[/]	62.0	-2.0	52.0

89	[*]	62.0	-62.0	0.0	149	[*]	58.0	-82.0	0.0
90	[/]	98.0	-62.0	0.0	150	[/]	58.0	-82.0	52.0
91	[/]	98.0	-98.0	0.0	151	[*]	58.0	-98.0	0.0
92	[/]	62.0	-98.0	0.0	152	[/]	58.0	-98.0	52.0
93	[/]	62.0	-62.0	0.0	153	[*]	-18.0	-98.0	0.0
94	[*]	62.0	-62.0	52.0	154	[/]	-18.0	-98.0	52.0
95	[/]	98.0	-62.0	52.0	155	[*]	-18.0	-22.0	0.0
96	[/]	98.0	-98.0	52.0	156	[/]	-18.0	-22.0	52.0
97	[/]	62.0	-98.0	52.0	157	[*]	-60.0	100.0	0.0
98	[/]	62.0	-62.0	52.0	158	[/]	-80.0	100.0	0.0
99	[*]	62.0	-62.0	0.0	159	[/]	-80.0	40.0	0.0
100	[/]	62.0	-62.0	52.0	160	[*]	-60.0	100.0	52.0
101	[*]	98.0	-62.0	0.0	161	[/]	-80.0	100.0	52.0
102	[/]	98.0	-62.0	52.0	162	[/]	-80.0	40.0	52.0
103	[*]	98.0	-98.0	0.0	163	[*]	-60.0	100.0	0.0
104	[/]	98.0	-98.0	52.0	164	[/]	-60.0	100.0	52.0
105	[*]	62.0	-98.0	0.0	165	[*]	-80.0	100.0	0.0
106	[/]	62.0	-98.0	52.0	166	[/]	-80.0	100.0	52.0
107	[*]	62.0	-62.0	0.0	167	[*]	-80.0	40.0	0.0
108	[/]	62.0	-62.0	52.0	168	[/]	-80.0	40.0	52.0
109	[*]	-78.0	-22.0	0.0	169	[*]	-82.0	40.0	0.0
110	[/]	-22.0	-22.0	0.0	170	[/]	-78.0	40.0	0.0
111	[/]	-22.0	-98.0	0.0	171	[/]	-78.0	22.0	0.0
112	[/]	-78.0	-98.0	0.0	172	[/]	-38.0	22.0	0.0
113	[/]	-78.0	-22.0	0.0	173	[/]	-38.0	2.0	0.0
114	[*]	-78.0	-22.0	52.0	174	[/]	2.0	2.0	0.0
115	[/]	-22.0	-22.0	52.0	175	[/]	2.0	-2.0	0.0
116	[/]	-22.0	-98.0	52.0	176	[/]	-42.0	-2.0	0.0
117	[/]	-78.0	-98.0	52.0	177	[/]	-42.0	18.0	0.0
118	[/]	-78.0	-22.0	52.0	178	[/]	-82.0	18.0	0.0
119	[*]	-78.0	-22.0	0.0	179	[/]	-82.0	40.0	0.0
120	[/]	-78.0	-22.0	52.0	180	[*]	-82.0	40.0	52.0
121	[*]	-22.0	-22.0	0.0	181	[/]	-78.0	40.0	52.0
122	[/]	-22.0	-22.0	52.0	182	[/]	-78.0	22.0	52.0
123	[*]	-22.0	-98.0	0.0	183	[/]	-38.0	22.0	52.0
124	[/]	-22.0	-98.0	52.0	184	[/]	-38.0	2.0	52.0
125	[*]	-78.0	-98.0	0.0	185	[/]	2.0	2.0	52.0
126	[/]	-78.0	-98.0	52.0	186	[/]	2.0	-2.0	52.0
127	[*]	-78.0	-22.0	0.0	187	[/]	-42.0	-2.0	52.0
128	[/]	-78.0	-22.0	52.0	188	[/]	-42.0	18.0	52.0
129	[*]	-18.0	-22.0	0.0	189	[/]	-82.0	18.0	52.0
130	[/]	38.0	-22.0	0.0	190	[/]	-82.0	40.0	52.0
131	[/]	38.0	-82.0	0.0	191	[*]	-82.0	40.0	0.0
132	[/]	58.0	-82.0	0.0	192	[/]	-82.0	40.0	52.0
133	[/]	58.0	-98.0	0.0	193	[*]	-78.0	40.0	0.0
134	[/]	-18.0	-98.0	0.0	194	[/]	-78.0	40.0	52.0
135	[/]	-18.0	-22.0	0.0	195	[*]	-78.0	22.0	0.0
136	[*]	-18.0	-22.0	52.0	196	[/]	-78.0	22.0	52.0
137	[/]	38.0	-22.0	52.0	197	[*]	-38.0	22.0	0.0
138	[/]	38.0	-82.0	52.0	198	[/]	-38.0	22.0	52.0
139	[/]	58.0	-82.0	52.0	199	[*]	-38.0	2.0	0.0
140	[/]	58.0	-98.0	52.0	200	[/]	-38.0	2.0	52.0
141	[/]	-18.0	-98.0	52.0	201	[*]	2.0	2.0	0.0
142	[/]	-18.0	-22.0	52.0	202	[/]	2.0	2.0	52.0
143	[*]	-18.0	-22.0	0.0	203	[*]	2.0	-2.0	0.0
144	[/]	-18.0	-22.0	52.0	204	[/]	2.0	-2.0	52.0
145	[*]	38.0	-22.0	0.0	205	[*]	-42.0	-2.0	0.0
146	[/]	38.0	-22.0	52.0	206	[/]	-42.0	-2.0	52.0
147	[*]	38.0	-82.0	0.0	207	[*]	-42.0	18.0	0.0
148	[/]	38.0	-82.0	52.0	208	[/]	-42.0	18.0	52.0

```
209    [*]   -82.0    18.0     0.0
210    [/]   -82.0    18.0    52.0
211    [*]   -82.0    40.0     0.0
212    [/]   -82.0    40.0    52.0
213    [*]   -72.0    18.0    44.0
214    [/]   -68.0    18.0    44.0
215    [/]   -68.0   -18.0    44.0
216    [/]   -72.0   -18.0    44.0
217    [/]   -72.0    18.0    44.0
218    [*]   -72.0    18.0    52.0
219    [/]   -68.0    18.0    52.0
220    [/]   -68.0   -18.0    52.0
221    [/]   -72.0   -18.0    52.0
222    [/]   -72.0    18.0    52.0
223    [*]   -72.0    18.0    44.0
224    [/]   -72.0    18.0    52.0
225    [*]   -68.0    18.0    44.0
226    [/]   -68.0    18.0    52.0
227    [*]   -68.0   -18.0    44.0
228    [/]   -68.0   -18.0    52.0
229    [*]   -72,0   -18.0    44.0
230    [/]   -72.0   -18.0    52.0
231    [*]   -40.0   100.0     0.0
232    [/]    -2.0   100.0     0.0
233    [*]   -40.0   100.0    40.0
234    [/]    -2.0   100.0    40.0
235    [*]   -40.0   100.0     0.0
236    [/]   -40.0   100.0    40.0
237    [*]    -2.0   100.0     0.0
238    [/]    -2.0   100.0    40.0
239    [*]    20.0   100.0    20.0
240    [/]    58.0   100.0    20.0
241    [*]    20.0   100.0    40.0
242    [/]    58.0   100.0    40.0
243    [*]    20.0   100.0    20.0
244    [/]    20.0   100.0    40.0
245    [*]    58.0   100.0    20.0
246    [/]    58.0   100.0    40.0
247    [*]    60.0    98.0    20.0
248    [/]    60.0    40.0    20.0
249    [*]    60.0    98.0    40.0
250    [/]    60.0    40.0    40.0
251    [*]    60.0    98.0    20.0
252    [/]    60.0    98.0    40.0
253    [*]    60.0    40.0    20.0
254    [/]    60.0    40.0    40.0
255    [*]    60.0    16.0     0.0
256    [/]    60.0     2.0     0.0
257    [*]    60.0    16.0    40.0
258    [/]    60.0     2.0    40.0
259    [*]    60.0    16.0     0.0
260    [/]    60.0    16.0    40.0
261    [*]    60.0     2.0     0.0
262    [/]    60.0     2.0    40.0
263    [*]   -80.0   -40.0     0.0
264    [/]   -80.0   -80.0     0.0
265    [*]   -80.0   -40.0    40.0
266    [/]   -80.0   -80.0    40.0
267    [*]   -80.0   -40.0     0.0
268    [/]   -80.0   -40.0    40.0
269    [*]   -80.0   -80.0     0.0
270    [/]   -80.0   -80.0    40.0
271    [*]   -60.0  -100.0     0.0
272    [/]   -22.0  -100.0     0.0
273    [*]   -60.0  -100.0    40.0
274    [/]   -22.0  -100.0    40.0
275    [*]   -60.0  -100.0     0.0
276    [/]   -60.0  -100.0    40.0
277    [*]   -22.0  -100.0     0.0
278    [/]   -22.0  -100.0    40.0
279    [*]    40.0   -82.0     0.0
280    [/]    40.0   -98.0     0.0
281    [*]    40.0   -82.0    52.0
282    [/]    40.0   -98.0    52.0
283    [*]    40.0   -82.0     0.0
284    [/]    40.0   -82.0    52.0
285    [*]    40.0   -98.0     0.0
286    [/]    40.0   -98.0    52.0
287    [*]   -18.0  -100.0     0.0
288    [/]    38.0  -100.0     0.0
289    [*]   -18.0  -100.0    40.0
290    [/]    38.0  -100.0    40.0
291    [*]   -18.0  -100.0     0.0
292    [/]   -18.0  -100.0    40.0
293    [*]    38.0  -100.0     0.0
294    [/]    38.0  -100.0    40.0
295    [*]   -40.0   -20.0     0.0
296    [/]   -22.0   -20.0     0.0
297    [*]   -40.0   -20.0    40.0
298    [/]   -22.0   -20.0    40.0
299    [*]   -40.0   -20.0     0.0
300    [/]   -40.0   -20.0    40.0
301    [*]   -22.0   -20.0     0.0
302    [/]   -22.0   -20.0    40.0
303    [*]   -18.0   -20.0     0.0
304    [/]     0.0   -20.0     0.0
305    [*]   -18.0   -20.0    40.0
306    [/]     0.0   -20.0    40.0
307    [*]   -18.0   -20.0     0.0
308    [/]   -18.0   -20.0    40.0
309    [*]     0.0   -20.0     0.0
310    [/]     0.0   -20.0    40.0
311    [*]    60.0   -40.0     0.0
312    [/]    60.0   -58.0     0.0
313    [*]    60.0   -40.0    40.0
314    [/]    60.0   -58.0    40.0
315    [*]    60.0   -40.0     0.0
316    [/]    60.0   -40.0    40.0
317    [*]    60.0   -58.0     0.0
318    [/]    60.0   -58.0    40.0
319    [*]    60.0   -62.0     0.0
320    [/]    60.0   -78.0     0.0
321    [*]    60.0   -62.0    40.0
322    [/]    60.0   -78.0    40.0
323    [*]    60.0   -62.0     0.0
324    [/]    60.0   -62.0    40.0
325    [*]    60.0   -78.0     0.0
326    [/]    60.0   -78.0    40.0
```

327	[*]	44.0	98.0	0.0	385	[/]	58.0	40.0	16.0
328	[/]	58.0	98.0	0.0	386	[*]	58.0	24.0	0.0
329	[/]	58.0	40.0	0.0	387	[/]	58.0	24.0	16.0
330	[/]	44.0	40.0	0.0	388	[*]	12.0	24.0	0.0
331	[/]	44.0	98.0	0.0	389	[/]	12.0	24.0	16.0
332	[*]	44.0	98.0	16.0	390	[*]	12.0	40.0	0.0
333	[/]	58.0	98.0	16.0	391	[/]	12.0	40.0	16.0
334	[/]	58.0	40.0	16.0	392	[*]	44.0	98.0	44.0
335	[/]	44.0	40.0	16.0	393	[/]	58.0	98.0	44.0
336	[/]	44.0	98.0	16.0	394	[/]	58.0	24.0	44.0
337	[*]	44.0	98.0	0.0	395	[/]	12.0	24.0	44.0
338	[/]	44.0	98.0	16.0	396	[/]	12.0	40.0	44.0
339	[*]	58.0	98.0	0.0	397	[/]	44.0	40.0	44.0
340	[/]	58.0	98.0	16.0	398	[/]	44.0	98.0	44.0
341	[*]	58.0	40.0	0.0	399	[*]	44.0	98.0	52.0
342	[/]	58.0	40.0	16.0	400	[/]	58.0	98.0	52.0
343	[*]	44.0	40.0	0.0	401	[/]	58.0	24.0	52.0
344	[/]	44.0	40.0	16.0	402	[/]	12.0	24.0	52.0
345	[*]	44.0	98.0	0.0	403	[/]	12.0	40.0	52.0
346	[/]	44.0	98.0	16.0	404	[/]	44.0	40.0	52.0
347	[*]	46.0	80.0	12.0	405	[/]	44.0	98.0	52.0
348	[/]	56.0	80.0	12.0	406	[*]	44.0	98.0	44.0
349	[/]	56.0	60.0	12.0	407	[/]	44.0	98.0	52.0
350	[/]	46.0	60.0	12.0	408	[*]	58.0	98.0	44.0
351	[/]	46.0	80.0	12.0	409	[/]	58.0	98.0	52.0
352	[*]	46.0	80.0	16.0	410	[*]	58.0	24.0	44.0
353	[/]	56.0	80.0	16.0	411	[/]	58.0	24.0	52.0
354	[/]	56.0	60.0	16.0	412	[*]	12.0	24.0	44.0
355	[/]	46.0	60.0	16.0	413	[/]	12.0	24.0	52.0
356	[/]	46.0	80.0	16.0	414	[*]	12.0	40.0	44.0
357	[*]	46.0	80.0	12.0	415	[/]	12.0	40.0	52.0
358	[/]	46.0	80.0	16.0	416	[*]	44.0	40.0	44.0
359	[*]	56.0	80.0	12.0	417	[/]	44.0	40.0	52.0
360	[/]	56.0	80.0	16.0	418	[*]	44.0	98.0	44.0
361	[*]	56.0	60.0	12.0	419	[/]	44.0	98.0	52.0
362	[/]	56.0	60.0	16.0	420	[*]	-10.0	60.0	0.0
363	[*]	46.0	60.0	12.0	421	[/]	-25.0	86.0	0.0
364	[/]	46.0	60.0	16.0	422	[/]	-55.0	86.0	0.0
365	[*]	46.0	80.0	12.0	423	[/]	-70.0	60.0	0.0
366	[/]	46.0	80.0	16.0	424	[/]	-55.0	34.0	0.0
367	[*]	46.0	56.0	16.0	425	[/]	-25.0	34.0	0.0
368	[/]	56.0	56.0	16.0	426	[/]	-10.0	60.0	0.0
369	[/]	56.0	44.0	16.0	427	[*]	-78.0	-60.0	0.0
370	[/]	46.0	44.0	16.0	428	[/]	-22.0	-60.0	0.0
371	[/]	46.0	56.0	16.0	429	[*]	-60.0	-40.0	0.0
372	[*]	12.0	40.0	0.0	430	[/]	-22.0	-40.0	0.0
373	[/]	58.0	40.0	0.0	431	[*]	-78.0	-80.0	0.0
374	[/]	58.0	24.0	0.0	432	[/]	-40.0	-80.0	0.0
375	[/]	12.0	24.0	0.0	433	[*]	-60.0	-22.0	0.0
376	[/]	12.0	40.0	0.0	434	[/]	-60.0	-60.0	0.0
377	[*]	12.0	40.0	16.0	435	[*]	-40.0	-60.0	0.0
378	[/]	58.0	40.0	16.0	436	[/]	-40.0	-98.0	0.0
379	[/]	58.0	24.0	16.0	437	[*]	-71.0	18.0	0.0
380	[/]	12.0	24.0	16.0	438	[/]	-69.0	18.0	0.0
381	[/]	12.0	40.0	16.0	439	[/]	-69.0	-18.0	0.0
382	[*]	12.0	40.0	0.0	440	[/]	-71.0	-18.0	0.0
383	[/]	12.0	40.0	16.0	441	[/]	-71.0	18.0	0.0
384	[*]	58.0	40.0	0.0	442	[*]	-71.0	18.0	44.0

```
443     [/]   -69.0     18.0     44.0
444     [/]   -69.0    -18.0     44.0
445     [/]   -71.0    -18.0     44.0
446     [/]   -71.0     18.0     44.0
447     [*]   -71.0     18.0      0.0
448     [/]   -71.0     18.0     44.0
449     [*]   -69.0     18.0      0.0
450     [/]   -69.0     18.0     44.0
451     [*]   -69.0    -18.0      0.0
452     [/]   -69.0    -18.0     44.0
453     [*]   -71.0    -18.0      0.0
454     [/]   -71.0    -18.0     44.0
455     [*]   -71.0     18.0      0.0
456     [/]   -71.0     18.0     44.0
457     [*]   -70.0      6.0      0.0
458     [/]   -70.0    -18.0      0.0
459     [*]   -70.0      6.0     42.0
460     [/]   -70.0    -18.0     42.0
461     [*]   -70.0      6.0      0.0
462     [/]   -70.0      6.0     42.0
463     [*]   -70.0    -18.0      0.0
464     [/]   -70.0    -18.0     42.0
```

◆F-BASIC活用ソフト　　（山海堂発行）

ワープロとグラフ　（FM-7・NEW7・77）

潮　守著／B5判／2,100円

FM-7・NEW7・77で使えるプログラムのリストとその使い方・操作法を収載.

- 辞書付簡易日本語ワープロ. 縦印刷もできます。
- ビジネスグラフと統計表・統計グラフ. 複合グラフも表示できます.
- ディスクユーティリティ. アクセスのスピードをアップするフォーマットなど8種.
- FM⟷PCファイル変換. FMとPC間で相互にプログラムを読み込めます.
- そのほか, グラフィックパネル・金利計算・オルガンなど計11本収載.

フロッピーディスク頒布サービス

全プログラム17本をディスクで頒布しております. 5インチ, 3.5インチとも7000円です(送料当社負担). サイズを明記のうえ現金書留にてお申し込み下さい.

申し込み先　東京都文京区本郷　5-5-18　山海堂　営業部　MS係

本書を購入された読者が使用する場合を除き，本書収載のプログラムの一部または全部を（株）山海堂の許諾を得ないで，いかなる方法においても複写・複製したり，他に転載・転用することを禁じます。
操作上の質問は具体的に記入して「FMグラフィックス」係宛お願いします。
なお，入力ミスや改造による質問は一切お断わり致します。

FM実践グラフィックス

—— 基礎から3次元C.Gまで ——

〔定価はカバーに表示しています〕

昭和60年1月30日　初版第1刷発行
昭和62年4月30日　初版第3刷発行

著者　田島清悟
発行者　川井正男
印刷所　壮光舎印刷株式会社

発行所　株式会社　山海堂
〒113 東京都文京区本郷5-5-18
振替東京　4-194982番
営業直通電話　03-816-1617～9

© 1985 Tajima

落丁本・乱丁本はお取り替えいたします。

ISBN 4-381-09007-1

◆わかる・楽しい・役に立つ

山海堂発行

FM-7 プログラミング入門

高橋三雄 著／B5判　240頁／2300円

はじめてプログラムを作ってみようとするユーザーを対象に，旅費計算や野球選手の打率計算など，身近にある題材をとりあげて，考え方，組み立て方，命令の意味・使い方等をわかりやすく説明します。読者はしらずしらずのうちにプログラム作りになじめるように工夫されています。巻末には，ラリーゲーム，お絵かき，音楽，スロットマシン，成績処理，家計簿，タテ横集計，グラフ作成など15本のプログラムを収載しています。

（NEW7/77にも使えます）

◆ビジネスマンのための

山海堂発行

パソコン入門術

システム科学研究所 編／B5変形　210頁／1500円

現代はニューメディアの時代，情報化社会……と呼ばれる新しい時代に移行しつつあります。このニューエイジに生きるビジネスマンが最も強い関心を持っているものの一つにパソコンがあります。しかし，これまでは，パソコンへの対応に間違いはなかったでしょうか？　対応の仕方には，おかれた立場により，人それぞれに違いがあって当然ではないでしょうか。この本は新しい視点に立って，人それぞれの対応の仕方を考える手がかりを与えてくれるでしょう。

- パソコンに取り組むための新しい発想
- パソコンの仕組みと働き
- すすむOAとメカトロニクス
- パソコンを操作するには
- 10語でルンルンプログラミング
- おもな周辺機器のはなし
- ミニミニ用語じてん

◆対話形式による

山海堂発行

FM-7 わくわく活用入門

青木　徹 著／B5判　226頁／2100円

入門の段階から抜け出たいと思っているユーザーを対象にした本。漢字の簡単な入力法，表の作り方，並べ替え，検索法，中間色の出し方，ミュージックの楽しみ方，画面のセーブ・ロードルーチン，部分スクロールルーチン，クイックソートルーチン，数値編集ルーチンなどの各種ルーチン，グラフィックシンボルエディタプログラム，スロットマシンゲームプログラムなど，内容豊富な12章です。

（NEW7/77にも使えます）

◆BASICから入る

山海堂発行

FMマシン語入門の手引き

鹿野哲郎著／B5判変形　200頁／ 1900円

興味津々のマシン語の世界に入ってみませんか．ＢＡＳＩＣもマシン語にホンヤクされて，はじめてコンピュータに理解されるのです．この本では，みなさんが知っているＢＡＳＩＣを使ってマシン語をいじることからスタートします．そうすれば意外と近づきやすいのです．雑誌のマシン語プログラムを打ち込んで使う時のチェックサムのプログラムや，簡易アセンブラのプログラム，マシン語を楽に入力できるプログラム等も載せてあります．これらを道具として使ってマシン語に親しんでみましょう．パソコンを骨までしゃぶる使い方もマシン語を使えばできるのです．この本で，そんなパソコンライフのスタートを切ってはどうでしょう．

◆FM-7・NEW7・77

山海堂発行

FM活用ソフト　1

システム科学研究所編／B5判　192頁／ 2100円

FM-7シリーズは，ゲームマシンとしてばかりでなく豊富な力を持っております．スモールビジネスにも立派に使用できるのです．本書では，給与計算，項目を自由に設定できる名簿管理，販売管理，在庫管理の4本のプログラムを収載し，詳細な操作方法を述べております．名簿管理は項目を自由に設定できますので個人ユーザーにも非常に便利なものです．

――フロッピーディスク頒布します――

5インチ　8000円　　3.5インチ　8500円

詳細は本の目次前をご覧ください．

◆実務に役立つフロッピーの使い方

山海堂発行

FMディスクファイル活用法

竹本篤郎著／B5判　176頁／ 1800円

フロッピーディスクを活用することによってパソコンライフは大きく広がります．処理スピード，扱えるデータ量からいってもフロッピーなしにはパソコンライフは考えられない段階に来ているといえるでしょう．本書は顧客管理，単価表等のプログラム例を使って，シーケンシャルファイル，ランダムファイルにおけるデータの書き込み，読み出し，訂正，削除等の使い方をていねいに説明しております．また，ディスク空間を無駄なく使うための方法として，インデックスファイルの作り方，使い方も述べております．各章とも基礎から応用にわたって入門者にも理解できるように工夫された好入門書です．

◆F-BASIC活用ソフト　(FM-7・NEW7・77)

ワープロとグラフ

潮　守著／B5判／2,100円

- 辞書付日本語ワープロ
- ビジネスグラフ
- 統計表と統計グラフ
- ディスクユーティリティ
- FM⟷PC変換プログラム　ほか

※ディスク頒布サービスあります.

◆わかる・楽しい・役に立つ (FM-7・NEW7・77)

FM-7プログラミング入門

高橋三雄著／B5判／2,300円

お遊び，家庭，仕事などの身近にある題材をとり上げて，プログラムを作る時の考え方，作り方，発展のさせ方等をやさしく説明．巻末にプログラム集付き．

◆対話形式による　(FM-7・NEW7・77)

FM-7 わくわく活用入門

青木　微著／B5判／2,100円

- グラフィックスを楽しもう
- 表を作ってみよう
- 漢字の使い方
- 簡易ソフトVIPの活用法
- 活用テクニックルーチン
- 便利な付録・楽しい付録

◆基礎から3次元CGまで　(FM-7・NEW7・77)

FM実践グラフィックス

田島清悟著／B5変形／1,800円

グラフィックの基礎から始め，むずかしい数学などは使わないで3次元を実践する．カンドコロを押さえて平易に解説．ディスク頒布サービスあります．

◆F-BASIC　(FM-7・NEW7・77)

ゲーム作りのノーハウ

ナラシノマイコンクラブ著／B5／2,300円

BASICだけを使って，ゲーム作りのテクニックやノーハウを明らかにします．パターンジェネレータ，サウンドジェネレータのほかサンプルゲームのリストを収載

◆ビジネスに役立つ　(FM-7・NEW7・77)

FM活用ソフト 1

B5判　192頁／2100円

スモールビジネス用にプログラムリストを収載し，その使用法を述べます．給与計算・販売管理・在庫管理・会員名簿管理など．

◆実務に役立つ　(FM-7・NEW7・77)

FMディスクファイルの活用法

B5判　176頁

ディスクを使うことによってパソコン活用の範囲は大きく広がります．実務に役立つ使い方の基礎から応用までを入門者にも分かるようていねいに解説しています．

◆BASICから入る　(FM-7・NEW7・77)

FMマシン語入門の手引き

B5変形　200頁／1900円

興味津々のマシン語を読者のなじんでいるBASICからアタック．初めてマシン語をかじってみようと思っている人，足踏みしている人へのおすすめ本です．

◆親子で楽しめる　(MZ-700・MZ1500)

MZ-700プログラミング入門

高橋三雄著／B5判／2,300円

各機種と互換性の高いHuBASICを使い，プログラムの作り方，並べ替えや検索の仕方，簡易言語HuCALの使い方を豊富な実例で解説．プログラム集付き．

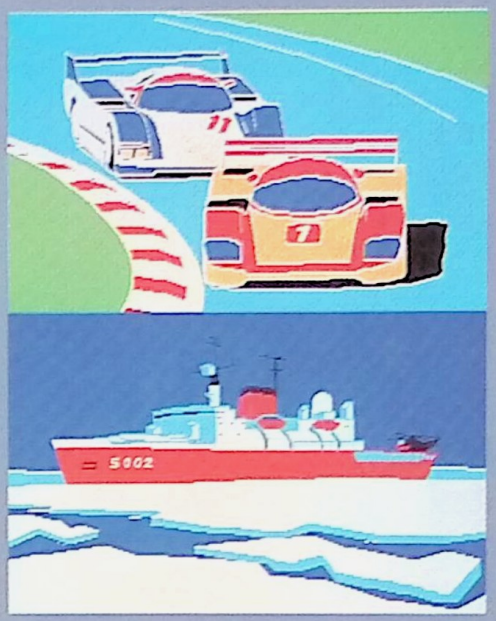
◆ビジネスマンのための

パソコン入門術

システム科学研究所／B5変形／1,500円

情報化社会におけるパソコンとの対応の仕方・使い方・簡単なプログラミング・周辺機器の話・操作法・OAとメカトロニクスのはなしなど満載

山海堂発行



FM実践グラフィックス

第1章
写真と図で綴るFMグラフィックス

第2章
基礎のグラフィックス

第3章
実践プログラミング

第4章
3次元グラフィックス

第5章
3次元実用プログラム3題

ISBN4-381-09007-1 C3055 ¥1800E

山海堂　定価 1800円